Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.084506

ARTICLE

Multimodal Implicit Representation Steganography Based on Point Cloud Representation

Yuwei Lu[1,2], **Jia Liu**[1,2,*], **Qiya Wang**[1,2], **Yujie Liu**[1,2] **and Peng Luo**[1,2]

[1]College of Cryptography Engineering, Engineering University of PAP, Xi'an, China
[2]Key Laboratory of Network and Information Security of PAP, Engineering University of PAP, Xi'an, China

*Corresponding Author: Jia Liu. Email: liujia1022@gmail.com

Received: 23 April 2026; Accepted: 22 June 2026; Published: 13 August 2026

ABSTRACT: Existing deep-learning-based steganography methods are typically designed for single-modality cover data and often rely on modality-specific network structures, which limits their cross-modal adaptability. To address this limitation, this paper proposes a multimodal implicit neural representation (INR) steganographic framework based on a point-cloud intermediate representation. The framework first fits the cover data as a carrier INR and samples the fitted carrier into a noisy point cloud. A pre-shared noise seed and secret key are then used to reproduce the carrier-derived point cloud and select a key-dependent point subset as the secret point cloud. Finally, a separate extractor, which is architecturally independent of the carrier INR, is trained to reconstruct the secret image from the secret point cloud. Instead of being treated as a decoder attached to the carrier network, the extractor can be encapsulated as a submodule within another neural network for delivery. On the receiver side, the secret image can be recovered only when the carrier INR, noise seed, secret key, and corresponding extractor are jointly available. Experimental results show that the reconstructed secret images achieve peak signal-to-noise ratio (PSNR) values above 40 dB on the CelebFaces Attributes-High Quality (CelebA-HQ), Common Objects in Context (COCO), and DIVerse 2K resolution (DIV2K) datasets.

KEYWORDS: Implicit neural representation (INR); INR-based steganography; steganography; information hiding; multimodal learning; point cloud

1 Introduction

Steganography is a technique for covert communication by concealing secret information within publicly transmitted cover media [1–4]. Deep-neural-network-based steganography has attracted increasing research attention because it can learn nonlinear embedding and extraction mappings from data [5–9]. These methods typically adopt an encoder–decoder architecture and leverage learned feature representations, improving embedding capacity, imperceptibility, and robustness. However, many image, video, and multi-image steganography methods [10–15] rely on discrete grid-based data representations, where the encoder and decoder are trained on pixels, frames, or modality-specific feature tensors. Consequently, such methods are often designed for particular data forms and face limitations when extended to heterogeneous cover data. Implicit neural representation (INR) [16,17] represents signals as continuous coordinate-to-feature mappings parameterized by neural networks, providing a flexible representation form for data of different modalities and resolutions. Recent INR-based steganography methods [18–22] use neural functions, network structures, or parameter spaces as carriers for secret information and implement hiding through function expansion, neuron pruning, weight replacement, or parameter allocation. These methods demonstrate the

feasibility of function-domain and parameter-domain information hiding. At the same time, they usually involve additional optimization or structural operations on the carrier INR, and the embedding strategy may still need to be adapted to different signal forms.

A central requirement for multimodal steganography is to construct a representation paradigm that is compatible with different cover modalities. As a coordinate-driven sampling form, a point cloud can organize sampled INR outputs into coordinate–feature point sets, thereby providing a common intermediate representation for cross-modal steganographic processing. In the proposed framework, the fitted carrier INR is not further modified after cover modeling. Instead, secret image recovery is achieved through point-cloud sampling, key-driven point subset selection, and a separate message extractor. Importantly, the carrier INR itself does not store or encode the secret message; the hiding process is defined by the key-controlled selection of point positions from the carrier-derived point cloud. The recovery of secret information is conditioned on the carrier INR, the noise seed, the secret key, and the corresponding extractor. Thus, the carrier INR provides a continuously samplable representation of the cover content, while the secret recovery process is defined by the relationship among the carrier-derived point cloud, the pre-shared noise seed, the secret key, and the message extractor.

Based on the above considerations, this paper proposes a point-cloud-based implicit neural representation steganographic framework. As shown in Fig. 1, the sender first models the cover data using an implicit neural representation and samples the fitted carrier INR to obtain a point-cloud representation. Subsequently, a pre-shared noise seed is used to make the point-cloud perturbation reproducible, and a secret key is used to select a subset from the carrier-derived point cloud as the secret point cloud. A message extractor is trained on this secret point cloud to recover the secret image. Finally, the trained carrier network and the network containing the extractor are transmitted through the public channel, where the extractor can be embedded within a neural network that performs other explicit tasks [23] for encapsulated delivery. On the receiver side, a public receiver can reconstruct the cover content through the carrier network, while an authorized receiver holding the correct noise seed and secret key can locate the secret point cloud and invoke the corresponding extraction module to recover the secret image.

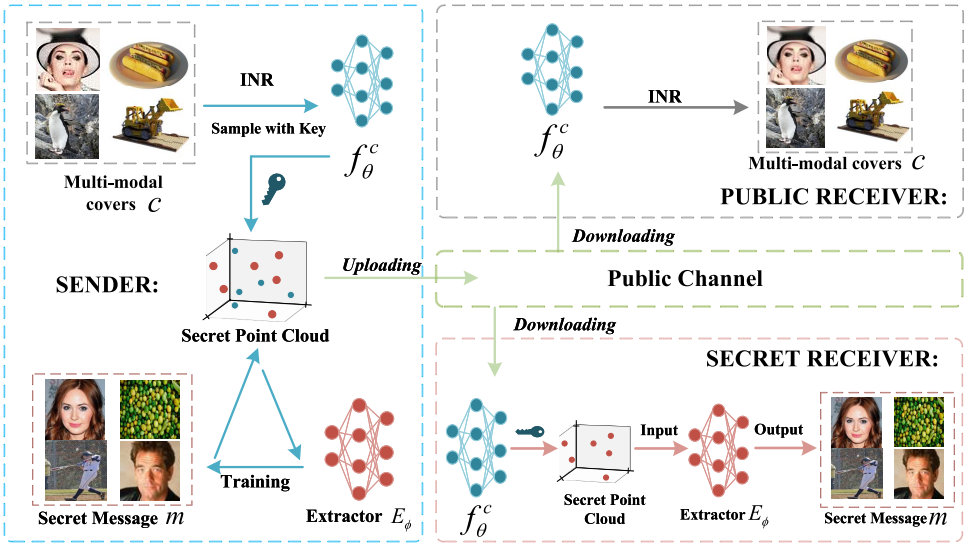

Figure 1: Communication scenario of the proposed point-cloud-based multimodal INR steganographic framework.

The main contributions of this paper are summarized as follows:

1. We propose a multimodal INR steganographic framework based on implicit neural representation, using coordinate–feature point clouds as an intermediate representation. This provides a unified data form for steganographic processing across different cover modalities and reduces the dependence on modality-specific grid representations.
2. We propose a key-driven random point-cloud selection mechanism. By introducing a key-controlled strategy, the method selects point positions from the carrier-derived point cloud to construct the secret point cloud. This mechanism defines the secret reconstruction condition at the point-cloud level and avoids direct modification of the fitted carrier INR parameters or structure.
3. We design a point-cloud-based message extractor that is architecturally separated from the carrier network and reconstructs the secret image from the key-selected point cloud. The extractor provides a unified extraction interface under the point-cloud representation.

2 Related Work

2.1 Deep Steganography

Deep steganography typically utilizes deep neural networks to construct encoder–decoder mapping relationships for embedding and extracting secret information. Baluja [5] proposed a convolutional-neural-network-based architecture for hiding a secret image within a cover image while maintaining the visual quality of the generated stego image. Hayes and Danezis [24] introduced adversarial training into the generation of steganographic images, and Zhu et al. [6] designed a deep hiding framework with differentiable noise layers to simulate channel distortions and improve robustness under image transformations. In terms of generative models, Zhang et al. [9] proposed SteganoGAN, a generative adversarial network (GAN)-based image steganography method for high-capacity message embedding. Targeting video data, Weng et al. [10] exploited temporal correlations between video frames and designed a dual-branch network structure to process original frames and highly sparse inter-frame residuals, applying convolutional networks to video steganography. Furthermore, Jing et al. [7] introduced invertible neural networks (INNs) and proposed the HiNet framework, utilizing the invertibility of the network structure to recover both cover and secret images. Lu et al. [25] further explored large-capacity image steganography based on invertible neural networks. However, the aforementioned deep-learning-based methods mainly rely on discrete pixel grid representations. As the spatial or temporal resolution increases, the computational and memory costs of these models usually increase accordingly. In addition, such network architectures are often designed for specific data modalities, which makes it difficult to provide a unified data hiding solution for heterogeneous cover data. This motivates the use of a common intermediate representation for organizing cover data from different modalities.

2.2 INR Steganography

In recent years, neural networks have been increasingly used not only as data processing models but also as representations for multimedia signals. As a coordinate-based data representation paradigm, INR represents signals such as images, audio, video, and three-dimensional (3D) scenes as continuous mapping functions parameterized by neural networks [16,17]. This continuous representation reduces the dependence on discrete sampling grids and provides a neural function space for information hiding technologies. Recently, researchers have utilized the parameter space, functional form, or structural characteristics of INR models to achieve secret information embedding. In 3D scene representation, Neural Radiance Fields (NeRF) [17] have shown the ability of INRs to model complex scenes, and related studies have explored information embedding and copyright protection within neural radiance fields [26,27]. Luo et al. [18]

proposed function steganography based on INRs, which embeds the implicit function representing secret information into the structure of the cover function. Building on this line of research, Dong et al. [19] introduced a model pruning strategy. This scheme first trains an INR function representing the secret image by masking partial neurons and uses the location indices of critical neurons as the private key; subsequently, it freezes these parameters and uses the remaining neurons to fit the cover image. To further enhance embedding capacity, Dong et al. [20] proposed the StegaINR4MIH framework, which exploits parameter redundancy in deep neural networks and utilizes a weight-magnitude-based selection strategy to replace and embed parameters of multiple secret images into a single carrier network, achieving multi-image hiding. The aforementioned INR steganography schemes demonstrate the feasibility of information hiding in neural function or parameter spaces. These methods usually involve structural or parameter-level operations on the carrier INR, such as function expansion, neuron pruning, or weight replacement. In this paper, INR is used to represent the cover as a continuous function, and the fitted carrier function is sampled into a coordinate–feature point cloud. Secret image recovery is then formulated through key-driven secret point-cloud selection and a separately trained extractor, so that the fitted carrier INR does not need to be further modified after cover modeling.

2.3 Multimodal Steganography

Multimodal steganography [28–30] aims to perform information hiding across heterogeneous data forms, such as images, audio, video, text, and neural representations. Early deep-learning-based multimodal steganography methods mostly adopted end-to-end Convolutional Neural Network (CNN) architectures, directly implementing cross-modal information embedding within the pixel or feature space. Kishore et al. [31] proposed an audio-image steganography framework based on Deep Convolutional Neural Networks (DCNNs), embedding one-dimensional audio signals as secret information into two-dimensional images. This method uses the nonlinear mapping capability of CNNs to connect audio features with the image-domain hiding process. However, such methods are typically designed for specific modality pairs, such as audio-to-image hiding, and require separate training of encoding and extraction models for the corresponding data forms. In recent years, methods based on INR have provided a new paradigm for multimodal steganography. Luo et al. [18] proposed the StegaINR framework, designing a hybrid expansion strategy capable of embedding a secret function representing one modality, such as images, into the structure of a cover function representing another modality, such as 3D models or meteorological data. Data format unification is achieved in the function domain, showing the feasibility of multimodal steganography from a functional representation perspective. Song et al. proposed Unified Steganography via Implicit Neural Representation (U-INR) [22], utilizing the parameter space of INRs to achieve unified multimodal representation. Implicit Steganography Beyond the Constraints of Modality (INRSteg) [21] further explored the multimodal potential of INRs by adopting a parameter partition allocation strategy. After converting multimodal secret data into INR weights, it improves the imperceptibility of the parameter distribution through layer-wise permutation and supports the coexistence of multiple secret data. Han et al. [32] proposed a deep cross-modal steganography framework based on neural representations and used quantization-aware optimization to reduce the conversion error between neural weights and secret data. In this paper, point clouds are used as a unified intermediate data representation for multimodal INR steganography. The carrier data are first represented by INRs and then organized as coordinate–feature point clouds. The secret recovery process is defined at the point-cloud level through key-driven point subset selection and message extraction, which separates cover modeling from secret image reconstruction.

3 Proposed Method

3.1 Overall Framework

The complete workflow of the proposed scheme is shown in Fig. 2.

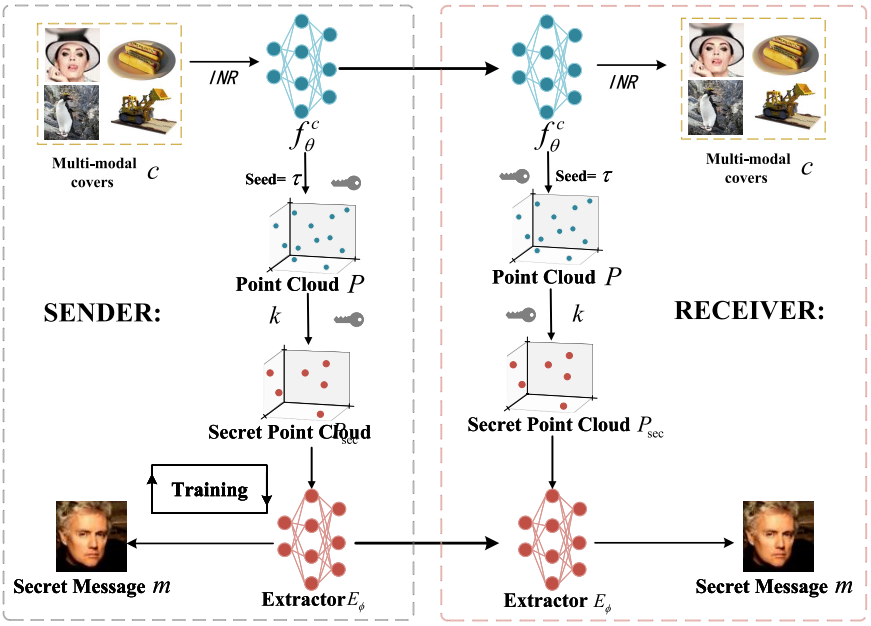

Figure 2: Flowchart of the multimodal INR steganography framework based on point-cloud intermediate representation.

Step 1: Implicit neural representation of the cover data. The sender represents the cover data c using an implicit neural representation. Let INR_p denote the implicit representation process. The neural function f_θ is trained on the coordinate–feature pairs of the cover data, and the fitted carrier function is denoted as f_θ^c. The implicit representation process of the cover data c can be written as

$$f_\theta^c = \mathrm{INR}_p(c, f_\theta). \tag{1}$$

Step 2: Point-cloud sampling of the carrier function. The carrier function f_θ^c is sampled into a point cloud P. During this process, U_{sam} denotes the sampled coordinate set, and τ is used as the random seed of the noise process to make the perturbation reproducible. The sampling process is formulated as

$$P = \mathrm{sample}(f_\theta^c, U_{\mathrm{sam}}, \tau). \tag{2}$$

Step 3: Secret point-cloud generation based on the key. The secret key k is used to generate the index set $I_{\mathrm{sam}}(k)$. The index set is then used to select points from the complete point cloud P and generate the secret point cloud P_s. This process is expressed as

$$P_s = P\big|_{I_{\mathrm{sam}}(k)} = \{p_j \mid j \in I_{\mathrm{sam}}(k)\}. \tag{3}$$

Step 4: Training of the secret information extractor. The secret information extractor E_ϕ is trained to reconstruct the secret image m from the secret point cloud P_s. The extractor parameter ϕ is optimized to obtain the trained parameter $\phi^\star$, as given by

$$\phi^{\star} = \arg\min_{\phi} L_{\text{ext}}(E_{\phi}(P_s), m). \tag{4}$$

After completing the training of the secret information extractor, the sender can embed the extractor $E_{\phi^{\star}}$ as a submodule into a neural network that performs other explicit tasks [23]. Finally, the sender publishes this network together with the implicit neural network of the cover data, f_{θ}^{c}, in a public channel. Public receivers can obtain the public carrier f_{θ}^{c} and recover the cover data c from it. Authorized receivers reconstruct the secret information m using the carrier f_{θ}^{c}, the pre-agreed noise seed τ, the secret key k, and the corresponding extractor module $E_{\phi^{\star}}$.

Step 5: Secret image reconstruction. For the authorized receiver, the carrier function f_{θ}^{c} is first sampled using the same coordinate sampling rule and the pre-agreed noise seed τ to reproduce the complete point cloud P. The secret key k is then used to obtain the secret point cloud P_s from P. The point cloud P_s is input into the secret information extractor $E_{\phi^{\star}}$ to obtain the reconstructed secret image m'. The specific process is formulated as

$$m' = E_{\phi^{\star}}(P_s). \tag{5}$$

3.2 Implicit Neural Representation of the Carrier

In this paper, a multilayer perceptron (MLP) network is adopted as the underlying network for the implicit neural representation [16,17]. The initially randomly generated MLP network is denoted as f_{θ}, where θ represents the weight parameters. The coordinate–feature pairs of the carrier data c are defined as (u_i, v_i), where u_i denotes an element in the coordinate space and v_i denotes the corresponding element in the feature space. Here, $i \in \{0, 1, 2, \ldots, N-1\}$, and N represents the total number of coordinate–feature pairs (u_i, v_i). The network f_{θ} is trained on the carrier data c. The optimization objective is formulated as

$$\theta_c = \arg\min_{\theta} \sum_{i=0}^{N-1} \| f_{\theta}(u_i) - v_i \|_2^2, \qquad f_{\theta}^{c} = f_{\theta_c}. \tag{6}$$

When the carrier data are images, $I[x, y]$ denotes the red-green-blue (RGB) value at the pixel coordinate (x, y). The network f_{θ} is trained to map pixel coordinates to RGB values. The corresponding optimization objective is formulated as

$$\theta_c = \arg\min_{\theta} \sum_{x,y} \| f_{\theta}(x, y) - I[x, y] \|_2^2, \qquad f_{\theta}^{c} = f_{\theta_c}. \tag{7}$$

3.3 Carrier Function Sampling into Point Clouds

To construct a discrete point-cloud domain for subsequent secret point-cloud selection, the continuous carrier implicit function f_{θ}^{c} is sampled into coordinate–feature point sets. This process consists of two steps: coordinate sampling and feature reconstruction.

First, sampling is performed in the coordinate space. Specifically, N discrete coordinate points are sampled from the coordinate space to form the sampled coordinate set $U_{\text{sam}} = \{u_i \mid i = 0, 1, \ldots, N-1\}$. The sampled coordinates are then input into the carrier implicit function to obtain the corresponding set of original feature values $V_{\text{sam}} = \{v_i = f_{\theta}^{c}(u_i) \mid u_i \in U_{\text{sam}}\}$. As shown in Fig. 3, noise is added to both the coordinates and the features to reduce regular grid patterns in the point cloud and to provide a reproducible perturbed point-cloud representation. Independent Gaussian noise is added to the coordinate u_i and the feature v_i, respectively. Let the noise seed be τ, the coordinate noise scale be σ_u, and the feature noise scale

be σ_v. The noise vectors satisfy $\varepsilon_{u,i} \sim \mathcal{N}(0, \sigma_u^2 I)$ and $\varepsilon_{v,i} \sim \mathcal{N}(0, \sigma_v^2 I)$. By using a fixed random seed τ, the noise distribution is made reproducible. The perturbed coordinate $\tilde{u}_i$ and feature $\tilde{v}_i$ are computed as follows:

$$\tilde{u}_i = u_i + \varepsilon_{u,i}, \qquad \tilde{v}_i = v_i + \varepsilon_{v,i}. \tag{8}$$

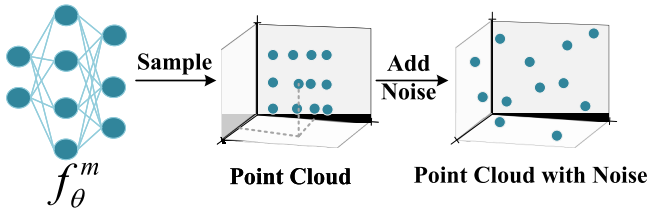

Figure 3: Schematic illustration of adding noise to the point cloud.

Finally, the perturbed coordinates and features are concatenated to form the noisy carrier point-cloud set:

$$P = \{(\tilde{u}_i, \tilde{v}_i) \mid i = 0, 1, \ldots, N-1\}. \tag{9}$$

Taking the case where the carrier data are an image c as an example, coordinate sampling is performed in the image coordinate space, and the sampled coordinate set is denoted as $U_{\text{sam}} = \{(x_i, y_i) \mid i = 0, 1, \ldots, N-1\}$. Then, color sampling is performed by substituting the sampled coordinates into the carrier implicit function, namely, $(r_i, g_i, b_i) = f_\theta^c(x_i, y_i)$. The original RGB color set is obtained as

$$RGB_{\text{sam}} = \{f_\theta^c(x_i, y_i) \mid (x_i, y_i) \in U_{\text{sam}}\} = \{(r_i, g_i, b_i) \mid i = 0, 1, \ldots, N-1\}. \tag{10}$$

For each coordinate point (x_i, y_i), a two-dimensional independent noise vector $\varepsilon_{u,i} = (\varepsilon_{u,i_1}, \varepsilon_{u,i_2})$ is generated. The perturbed coordinate $(\tilde{x}_i, \tilde{y}_i)$ is obtained by adding the noise to the original coordinate:

$$\tilde{x}_i = x_i + \varepsilon_{u,i_1}, \qquad \tilde{y}_i = y_i + \varepsilon_{u,i_2}. \tag{11}$$

Next, noise is added to the color feature. For the color (r_i, g_i, b_i) corresponding to each coordinate point, a three-dimensional independent noise vector $\varepsilon_{v,i} = (\varepsilon_{v,i_1}, \varepsilon_{v,i_2}, \varepsilon_{v,i_3})$ is generated. The perturbed color $(\tilde{r}_i, \tilde{g}_i, \tilde{b}_i)$ is obtained by adding the noise to the original color:

$$\tilde{r}_i = r_i + \varepsilon_{v,i_1}, \qquad \tilde{g}_i = g_i + \varepsilon_{v,i_2}, \qquad \tilde{b}_i = b_i + \varepsilon_{v,i_3}. \tag{12}$$

The noisy coordinate $(\tilde{x}_i, \tilde{y}_i)$ of each point is concatenated with its noisy color $(\tilde{r}_i, \tilde{g}_i, \tilde{b}_i)$ to form the noisy point cloud of the carrier. Let $P = \{p_i \mid i = 0, 1, \ldots, N-1\}$. The point cloud is specifically expressed as

$$P = \{(\tilde{x}_i, \tilde{y}_i, \tilde{r}_i, \tilde{g}_i, \tilde{b}_i) \mid i = 0, 1, \ldots, N-1\}. \tag{13}$$

3.4 Generation of the Secret Point Cloud

After generating the complete point cloud P, a subset is selected as the input data for the extractor. The subset selection process is controlled by the secret key k. From the full index set $I_{\text{full}} = \{0, 1, \ldots, N-1\}$, M non-repeated indices are selected using the key k to form the sampling index set $I_{\text{sam}}(k)$, which satisfies

$$I_{\text{sam}}(k) \subseteq I_{\text{full}}, \quad |I_{\text{sam}}(k)| = M, \quad i_a \neq i_b \text{ for } a \neq b. \tag{14}$$

The corresponding points are extracted from the complete point cloud P using the sampling index set $I_{\text{sam}}(k)$, thereby forming the secret point cloud $P_s = \{p_j \mid j \in I_{\text{sam}}(k)\}$. Its tensor representation is $P_s \in \mathbb{R}^{M \times d_p}$, where d_p denotes the dimension of each coordinate–feature point. For RGB image carriers, $d_p = 5$. The secret point cloud is expressed as

$$P_s = P\big|_{I_{\text{sam}}(k)} = \{p_j \mid j \in I_{\text{sam}}(k)\} = \{p_{i_0}, p_{i_1}, \ldots, p_{i_{M-1}}\}. \tag{15}$$

3.5 Training of the Secret Information Extractor

The role of the secret information extractor is to learn a secret reconstruction mapping conditioned on the point cloud selected by the key. Given the secret point cloud P_s, which is jointly generated by the correct carrier implicit neural representation, the noise seed, and the secret key, the extractor E_ϕ outputs the reconstructed secret image m'. By minimizing the reconstruction error, the model parameter ϕ is iteratively updated, and the trained extractor $E_{\phi^\star}$ is obtained, as shown in

$$\phi^\star = \arg\min_{\phi} \left\| E_\phi(P_s) - m \right\|_2^2. \tag{16}$$

To quantify the difference between the reconstruction result and the original secret image, the mean squared error is adopted as the loss function. The input is the sampled point cloud P_s, and the output is the reconstructed image m'. By optimizing the extractor parameter ϕ, the reconstructed image m' is made to approach the secret image m, namely $\phi^\star = \arg\min_\phi L(\phi)$, where $\phi^\star$ denotes the trained weight parameter of the extractor and $L(\phi)$ denotes the loss function. Let the secret image size be $H_m \times W_m$ with C channels. The loss function is formulated as

$$L(\phi) = \frac{1}{H_m W_m C} \sum_{x=0}^{H_m-1} \sum_{y=0}^{W_m-1} \sum_{\ell=1}^{C} \left(E_\phi(P_s)[x, y, \ell] - m[x, y, \ell] \right)^2. \tag{17}$$

Here, $E_\phi(P_s)[x, y, \ell]$ denotes the output pixel value of the extractor at coordinate (x, y) and channel ℓ, and $m[x, y, \ell]$ denotes the pixel value of the secret image at the corresponding position and channel. The secret point cloud P_s is jointly determined by f_θ^c, the noise seed τ, and the key k. If the input point cloud is generated using an incorrect key, an incorrect seed, another carrier, or a random distribution, the input point cloud does not match the training condition of the extractor, and the secret image cannot be reliably reconstructed.

3.6 Secret Information Extraction

In the secret extraction stage, the authorized receiver needs to simultaneously satisfy three conditions: obtaining the publicly released carrier implicit neural representation f_θ^c, possessing the correct random seed τ and secret key k, and invoking the corresponding extractor $E_{\phi^\star}$. During the secret information extraction process, after receiving the carrier function f_θ^c, the authorized receiver uses the carrier function f_θ^c together with the pre-agreed coordinate sampling rule, the noise seed τ, and the secret key k to reproduce the sender's secret point cloud, as shown in Eq. (18).

$$P_s = \text{sample}(f_\theta^c, U_{\text{sam}}, \tau)\big|_{I_{\text{sam}}(k)} = \{p_j \mid j \in I_{\text{sam}}(k)\}. \tag{18}$$

The receiver loads the trained parameter $\phi^\star$ of the extractor, inputs the secret point cloud P_s into the extractor $E_{\phi^\star}$, and reconstructs the secret image m', as shown in Eq. (19). If any of the above conditions is not satisfied, for example, if an incorrect key, an incorrect seed, or a point cloud from another carrier is used,

the input point cloud no longer matches the training condition of the extractor, and the secret image cannot be reliably reconstructed.

$$m' = E_{\phi^\star}\left(\text{sample}(f_\theta^c, U_{\text{sam}}, \tau)\big|_{I_{\text{sam}}(k)}\right). \tag{19}$$

4 Experiments and Result Analysis

4.1 Datasets and Experimental Settings

All experiments are implemented with PyTorch 1.7.0 and Python 3.8 on a server equipped with an NVIDIA GeForce RTX 2070 graphics processing unit (GPU) and Compute Unified Device Architecture (CUDA) 11.6. To evaluate the applicability of the proposed framework to different carrier sources, three types of carrier data are used, including natural images, 3D scenes, and meteorological grid data. Specifically, image carriers are selected from the CelebFaces Attributes-High Quality (CelebA-HQ) [33], Common Objects in Context (COCO) [34], and DIVerse 2K resolution (DIV2K) [35] datasets. For 3D scene data, data from the NeRF-Synthetic dataset [17] are used as carrier signals. For meteorological data, single-level temperature grid fields from the European Centre for Medium-Range Weather Forecasts Reanalysis version 5 (ERA5) dataset [36] are adopted, where the data from December 2022 are extracted and normalized for steganographic processing.

In the experimental design, the above three types of carrier data are first used to evaluate the applicability of the proposed framework under different carrier forms. Considering that image steganography provides more widely used comparison baselines, image data are then used as the main carrier type for the quantitative analysis of secret image quality, hiding capacity, robustness, and security. For each image dataset used in the image-carrier experiments, five carrier–secret image pairs are constructed, where each pair contains one cover image and one secret image. For each pair, the experiment is repeated three times using different random seeds. The reported quantitative results are the mean and standard deviation over the resulting measurements. For the multimodal carrier evaluation, five carrier–secret pairs are constructed for each carrier type and evaluated under the same repeated-trial protocol. Peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) [37] are adopted to evaluate image reconstruction quality and secret image recovery quality, while root mean square error (RMSE) and mean absolute error (MAE) are used for meteorological grid reconstruction.

For the network architecture and training parameters, the proposed framework consists of a carrier implicit representation network and a secret information extractor. For the carrier network, an MLP is used to fit the data. Taking RGB image carriers as an example, the network structure is set to $(2 \to 128 \to 128 \to 128 \to 3)$, where the input dimension is 2 for pixel coordinates and the output dimension is 3 for RGB values. The carrier network is trained for 2500 epochs using stochastic gradient descent with a learning rate of $\eta = 0.001$. For 3D and meteorological data, only the input layer size is adjusted according to the coordinate dimension, while the remaining structure is kept consistent.

The extractor adopts a shared MLP combined with global average pooling (AvgPool). Specifically, each coordinate–feature point is first fed into the same MLP to extract a 128-dimensional feature. Global average pooling is then performed over all point features to obtain a 128-dimensional global feature, which is finally passed through a fully connected (FC) layer to output the secret image. This process can be written as $m' = FC(AvgPool(SharedMLP(P_s)))$, where $SharedMLP$ extracts features from each point, $AvgPool$ aggregates all point features, and FC outputs the secret image. The extractor is trained for 150 epochs using stochastic gradient descent with a learning rate of $\eta = 0.0001$. The input dimension of the extractor is set according to the dimension of the coordinate–feature point cloud, and the output dimension is $H_m W_m C_m$, corresponding to the flattened secret image with height H_m, width W_m, and channel number C_m.

4.2 Multi-Modal Carriers

The point-cloud-based implicit neural representation can organize carrier data from different sources into coordinate–feature point-cloud representations. Experiments in this subsection are conducted on three carrier forms, including DIV2K image data, NeRF/3D scene data, and meteorological grid data. The image carriers are selected from the DIV2K [35] dataset. The 3D scene carriers are selected from the NeRF-Synthetic dataset [17]. During evaluation, a fixed rendered view is generated from each fitted 3D scene INR, and PSNR and SSIM are computed between the rendered view and the corresponding ground-truth view. The meteorological carriers are normalized single-level temperature grid fields from December 2022 in the ERA5 dataset [36], and their reconstruction accuracy is evaluated using RMSE and MAE.

In addition to basic multi-modal examples, we further consider diverse carrier settings, including complex natural images, different 3D scenes, and meteorological grid data. As shown in Fig. 4, representative qualitative results are presented. To further quantify the reconstruction performance under different carrier modalities, the carrier reconstruction quality and secret image recovery quality are reported in Table 1. As shown in Table 1, the proposed method can recover recognizable secret images under the tested carrier forms, including natural images, 3D scenes, and meteorological grid data. The results show that the point-cloud intermediate representation provides a consistent input form for the tested carrier modalities.

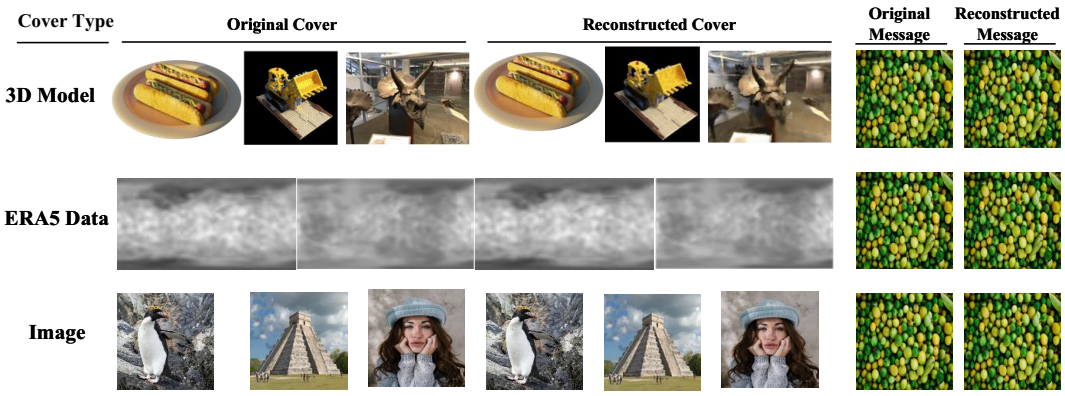

Figure 4: Qualitative results under different carrier types.

Table 1: Reconstruction quality under different carrier modalities.

Cover Data	Number of Pairs	Carrier Reconstruction Quality	Secret Image PSNR/SSIM
DIV2K	5	PSNR: 38.732 ± 1.106 dB; SSIM: 0.956 ± 0.012	48.013 ± 0.697/0.963 ± 0.001
NeRF-Synthetic	5	Test-view PSNR: 36.42 ± 1.32 dB; Test-view SSIM: 0.941 ± 0.021	44.736 ± 0.884/0.934 ± 0.002
ERA5	5	RMSE: 0.012 ± 0.003; MAE: 0.008 ± 0.002	45.218 ± 0.792/0.985 ± 0.002

Note: DIV2K denotes DIVerse 2K-resolution high-quality images. NeRF denotes neural radiance fields. ERA5 denotes the fifth-generation European Centre for Medium-Range Weather Forecasts atmospheric reanalysis. PSNR denotes peak signal-to-noise ratio. SSIM denotes structural similarity index measure. dB denotes decibel. RMSE denotes root mean squared error. MAE denotes mean absolute error.

4.3 Secret Image Quality

To evaluate the quality of secret image recovery, image-carrier experiments are conducted at a resolution of 128 × 128 on the CelebA-HQ [33], COCO [34], and DIV2K [35] datasets. As shown in Fig. 5, the first and second rows show the original and reconstructed cover images, while the third and fourth rows show the original and recovered secret images. The reconstructed cover images are obtained by sampling the public carrier INR, and the secret images are recovered by feeding the secret point cloud, reproduced using the agreed noise seed and secret key, into the corresponding extractor.

Figure 5: Cover reconstruction and secret image recovery results on different image datasets.

Table 2 reports the image quality results on different datasets, where the values are the mean and standard deviation over five carrier–secret image pairs and three repeated trials for each image dataset. PSNR, SSIM [37], RMSE, and MAE are used to evaluate the reconstructed cover images and recovered secret images. The recovered secret images obtain PSNR values above 46 dB and SSIM values between 0.963 and 0.989 on the tested image datasets. These results indicate that the carrier-extractor pipeline provides consistent secret image recovery quality under the 128 × 128 image-carrier setting.

Table 2: Image quality on different image datasets.

Dataset	Image Type	PSNR	SSIM	RMSE	MAE
CelebA-HQ	Reconstructed cover image	33.861 ± 1.214	0.753 ± 0.031	0.020 ± 0.003	0.015 ± 0.002
	Recovered secret image	47.529 ± 0.638	0.972 ± 0.001	0.003 ± 0.001	0.002 ± 0.001
COCO	Reconstructed cover image	37.632 ± 1.083	0.872 ± 0.026	0.013 ± 0.002	0.010 ± 0.001
	Recovered secret image	46.832 ± 0.714	0.989 ± 0.001	0.003 ± 0.001	0.001 ± 0.001
DIV2K	Reconstructed cover image	38.732 ± 1.106	0.956 ± 0.012	0.012 ± 0.002	0.009 ± 0.001
	Recovered secret image	48.013 ± 0.697	0.963 ± 0.001	0.002 ± 0.001	0.001 ± 0.001

Note: PSNR denotes peak signal-to-noise ratio, which measures image reconstruction fidelity in decibels, with higher values indicating better quality. SSIM denotes structural similarity index measure, which evaluates perceptual structural similarity between two images, with values closer to 1 indicating higher similarity. RMSE denotes root mean squared error, and MAE denotes mean absolute error; both measure pixel-wise reconstruction errors, where lower values indicate smaller distortion.

4.4 Steganographic Capacity and Transmission Cost Analysis

To analyze the capacity of the proposed method under different secret image scales, the cover image resolution is fixed at 128 × 128. Five groups of secret images with resolutions of 64 × 64, 128 × 128, 256 × 256, 512 × 512, and 1024 × 1024 are selected from the CelebA-HQ [33] dataset for training. We compare the visual difference between the original and reconstructed secret images, and quantitatively evaluate the nominal hiding capacity, recovery quality, and extractor-related transmission cost. PSNR and SSIM [37] are used as the main image-quality metrics.

Let the secret image size be $H_s \times W_s$ and the cover image resolution be $H_c \times W_c$. The effective secret payload is computed as $B_s = H_s \times W_s \times C \times q$, where C denotes the number of secret-image channels and q denotes the bit depth of each channel. The nominal capacity per cover pixel, measured in bits per pixel (bpp), is computed as $C_{\text{bpp}} = B_s/(H_c \times W_c)$. In this experiment, RGB secret images are used, with $C = 3$ and $q = 8$. Since the carrier INR is fixed in this analysis and the extractor E_ϕ is required for secret recovery, the extractor parameter number N_e is reported as a compact indicator of the resolution-dependent transmission cost. The capacity, extractor parameter number, and recovery quality are summarized in Table 3. As shown in Fig. 6, the first row shows the original secret images, and the second row shows the reconstructed secret images.

Table 3: Capacity, extractor parameters, and recovery quality under different secret image resolutions.

Resolution	Payload B_s/Mbit	C_{bpp}/bpp	N_e/M	PSNR/dB	SSIM
64 × 64	0.098	6	1.602	50.421 ± 0.621	0.989 ± 0.001
128 × 128	0.393	24	6.358	47.529 ± 0.638	0.972 ± 0.001
256 × 256	1.573	96	25.380	44.142 ± 0.816	0.951 ± 0.004
512 × 512	6.291	384	101.467	41.688 ± 0.952	0.892 ± 0.015
1024 × 1024	25.166	1536	405.816	40.132 ± 1.184	0.812 ± 0.026

Note: Mbit denotes megabits, and bpp denotes bits per pixel. PSNR/dB denotes the peak signal-to-noise ratio measured in decibels, and SSIM denotes the structural similarity index measure; higher values indicate better recovered image quality.

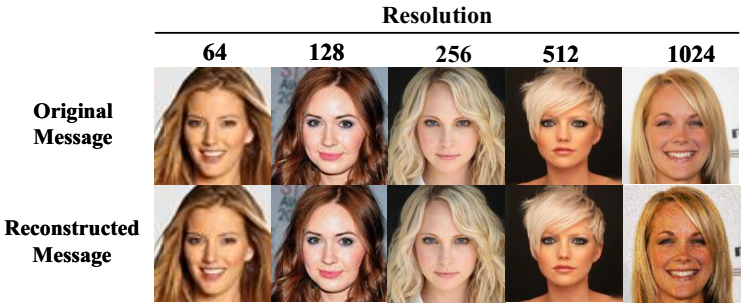

Figure 6: Image quality of secret images with different resolutions.

Fig. 6 and Table 3 show that, as the secret image resolution increases, the effective payload B_s, the nominal capacity per cover pixel C_{bpp}, and the extractor parameter number N_e all increase progressively. Within the range from 64 × 64 to 256 × 256, the recovered secret images maintain relatively high quality, with PSNR values higher than 44 dB and SSIM values close to or above 0.95. This indicates that, in this resolution range, the proposed framework can achieve a favorable balance among payload, recovery quality, and extractor scale. When the secret image resolution increases to 512 × 512 and 1024 × 1024, the payload B_s

further increases, but the extractor parameter number also grows significantly, and the structural similarity begins to decrease. In particular, under the 1024 × 1024 setting, the PSNR still remains above 40 dB, while the SSIM drops noticeably, indicating that structural recovery becomes more difficult under high-resolution secret reconstruction. In addition, the model-size-adjusted capacity after considering the extractor remains approximately within the range of 0.060–0.062 bits per parameter (bits/param), suggesting that the capacity expansion under the current architecture mainly comes with the increased output dimension and parameter scale of the extractor. Therefore, although the 512 × 512 and 1024 × 1024 settings provide higher C_{bpp}, they also introduce more evident extractor transmission cost and structural-fidelity degradation. Under the current implementation, these high-resolution settings are more suitable for analyzing the trend of capacity expansion rather than serving as preferred transmission configurations. Considering payload, recovery quality, and extractor scale together, the range from 64 × 64 to 256 × 256 can be regarded as a more balanced capacity setting.

4.5 Robustness

To evaluate robustness under parameter pruning, we apply L_1 unstructured pruning and L_n structured pruning to the carrier function. The pruning operation is applied only to the carrier function, while the extractor, noise seed, and key settings remain unchanged during evaluation. After pruning, the cover image is reconstructed from the pruned carrier function, and the secret image is recovered by feeding the corresponding secret point cloud into the original extractor.

Following the experimental setting in Section 4.1, the quantitative results are averaged over multiple cover–secret pairs and repeated trials. For qualitative visualization, one representative 128 × 128 cover–secret pair from the DIV2K [35] dataset is selected. The reconstructed cover and recovered secret images before and after pruning are compared with the original cover and secret images. As shown in Fig. 7, different pruning methods affect the recovery quality of both cover and secret images. The figure is organized into two pruning blocks: the left block corresponds to L_1 unstructured pruning, and the right block corresponds to L_n structured pruning. In each block, the rows correspond to the cover image and the secret image, respectively. The "Original" column shows the ground-truth cover and secret images, the column with pruning rate 0 shows the reconstructed cover and recovered secret results before pruning, and the remaining columns show the results after applying the corresponding pruning rates.

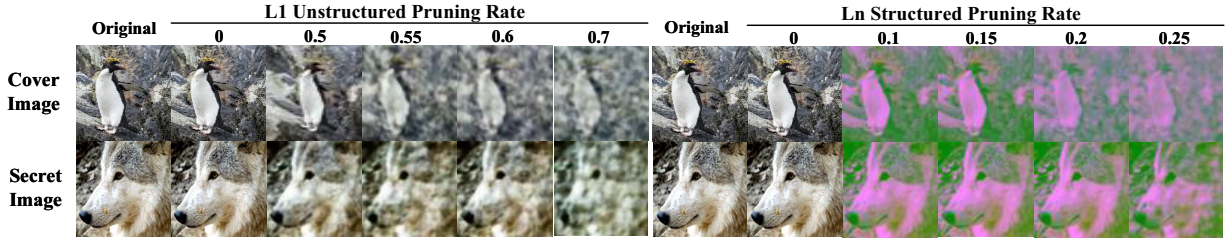

Figure 7: Cover and secret image results under different pruning methods.

Table 4 reports the PSNR values of the recovered cover and secret images under different pruning methods and pruning rates. The results show that the carrier-extractor pipeline is more stable under L_1 unstructured pruning than under L_n structured pruning. Under L_1 unstructured pruning, the secret image can still be recovered with a PSNR of 38.532 dB at a pruning rate of 0.5. When the pruning rate increases to 0.55 and above, the reconstruction quality decreases progressively. Under L_n structured pruning, the secret image achieves a PSNR of 27.018 dB at a pruning rate of 0.1, and the quality decreases more rapidly as the

pruning rate increases. These results indicate that the carrier-extractor pipeline retains a certain tolerance to L_1 unstructured pruning, while L_n structured pruning has a stronger influence on both cover reconstruction and secret recovery.

Table 4: PSNR values of images under different pruning settings.

Image Type	L_1 Unstructured Pruning Rate				L_n Structured Pruning Rate			
	0.5	**0.55**	**0.6**	**0.7**	**0.1**	**0.15**	**0.2**	**0.25**
Cover image	24.187 ± 1.104	20.743 ± 1.318	16.373 ± 1.506	9.813 ± 1.221	21.614 ± 1.285	18.367 ± 1.196	13.812 ± 1.418	8.057 ± 1.073
Secret image	38.532 ± 1.236	31.129 ± 1.442	20.092 ± 1.691	14.521 ± 1.284	27.018 ± 1.305	24.001 ± 1.418	16.532 ± 1.552	13.336 ± 1.297

4.6 Security Analysis

To evaluate the security of the proposed method, we conducted experiments from two aspects: steganalysis detectability and key sensitivity. The attacker is assumed to know the method pipeline and network structure and to have access to the publicly transmitted carrier INR. In the extractor-leakage setting, the attacker may also obtain the message extractor, but does not have the correct noise seed and secret key. The attacker's goals are to determine whether the public objects are associated with covert communication and to recover the secret image without the correct seed-key pair.

4.6.1 Steganalysis

The proposed method does not generate a conventional pixel-domain stego image. Therefore, detectability is evaluated from three objects: carrier INR parameters, carrier-reconstructed cover images, and candidate point subsets generated from the public carrier INR. Steganalysis is commonly formulated as a binary classification problem, and both hand-crafted statistical features and learning-based detectors are widely used in this task [2,38]. Following the support vector machine (SVM)-based function-parameter detector used in existing INR steganography [18], carrier INR parameters are first flattened and converted into normalized 50-bin weight histograms. The histogram features are combined with mean, variance, skewness, kurtosis, and L_2 norm, and polynomial-kernel and radial basis function (RBF)-kernel SVMs are used for binary classification. For carrier-reconstructed cover images, we extract hand-crafted image-domain statistical features and use an ensemble-style classifier. For candidate point subsets, we extract coordinate histograms, feature-value histograms, pairwise-distance statistics, and local density statistics, and use an SVM to distinguish key-conditioned point subsets from random point subsets with the same size.

All detection tasks are constructed as balanced binary classification problems. For the carrier-parameter and reconstructed-cover tests, positive samples are generated from carrier INRs used in the covert communication process, while negative samples are generated from ordinary carrier INRs trained with the same architecture and datasets. For the point-subset test, positive samples are key-conditioned candidate point subsets, while negative samples are randomly selected point subsets generated from the same carrier point clouds. The samples are divided into training, validation, and test sets at a ratio of 7:1:2, with Accuracy, area under the receiver operating characteristic curve (AUC), and F1-score as evaluation metrics.

As shown in Table 5, the Accuracy values of the three detection tasks remain close to 50%, while the AUC and F1-score values are also close to chance-level classification. These results indicate that the tested feature sets and classifiers do not provide stable discrimination between positive and negative samples in this setting.

Table 5: Steganalysis-based detectability evaluation.

Detection Object	Detector	Accuracy	AUC	F1-Score
Carrier INR parameters	50-bin weight-histogram SVM	50.8 ± 2.1%	0.512 ± 0.022	0.503 ± 0.025
Reconstructed cover image	Image-statistics classifier	52.1 ± 2.4%	0.529 ± 0.031	0.518 ± 0.028
Key-conditioned candidate point subset	Point-statistics SVM	51.6 ± 2.8%	0.521 ± 0.029	0.514 ± 0.030

Note: AUC denotes the area under the ROC curve. An AUC close to 0.5 indicates near-random detection, suggesting low detectability under the tested settings.

4.6.2 Key Security Analysis

We further analyze the secret image recovery quality under incorrect noise-seed and secret-key conditions. In this experiment, K_1 denotes the noise seed and K_2 denotes the point-selection key. The secret point cloud is jointly determined by the carrier INR, the noise seed K_1, and the secret key K_2. When any of these conditions is inconsistent with the extractor training condition, the generated point cloud deviates from the expected input distribution, and the secret image cannot be reliably recovered. Six settings are considered: correct K_1 and correct K_2, correct K_1 and wrong K_2, wrong K_1 and correct K_2, wrong K_1 and wrong K_2, random point-cloud input, and point clouds generated from another carrier. PSNR, SSIM, RMSE, and MAE are used to evaluate the difference between the recovered and original secret images. The results are shown in Table 6 and Fig. 8.

Table 6: Secret image recovery quality under different attack settings.

Attack Setting	PSNR/dB	SSIM	RMSE	MAE
Correct K_1 + correct K_2	48.013 ± 0.697	0.963 ± 0.001	0.002 ± 0.001	0.001 ± 0.001
Correct K_1 + wrong K_2	8.63 ± 0.82	0.023 ± 0.018	0.348 ± 0.027	0.272 ± 0.024
Wrong K_1 + correct K_2	11.64 ± 0.76	0.087 ± 0.026	0.281 ± 0.021	0.189 ± 0.019
Wrong K_1 + wrong K_2	9.38 ± 0.69	0.021 ± 0.015	0.362 ± 0.025	0.233 ± 0.023
Random point cloud	8.92 ± 0.73	0.018 ± 0.012	0.358 ± 0.029	0.293 ± 0.026
Point cloud from another carrier	10.21 ± 0.80	0.046 ± 0.021	0.309 ± 0.026	0.251 ± 0.022

Note: PSNR, SSIM, RMSE, and MAE denote peak signal-to-noise ratio, structural similarity index measure, root mean squared error, and mean absolute error, respectively. Higher PSNR/SSIM and lower RMSE/MAE indicate better image quality.

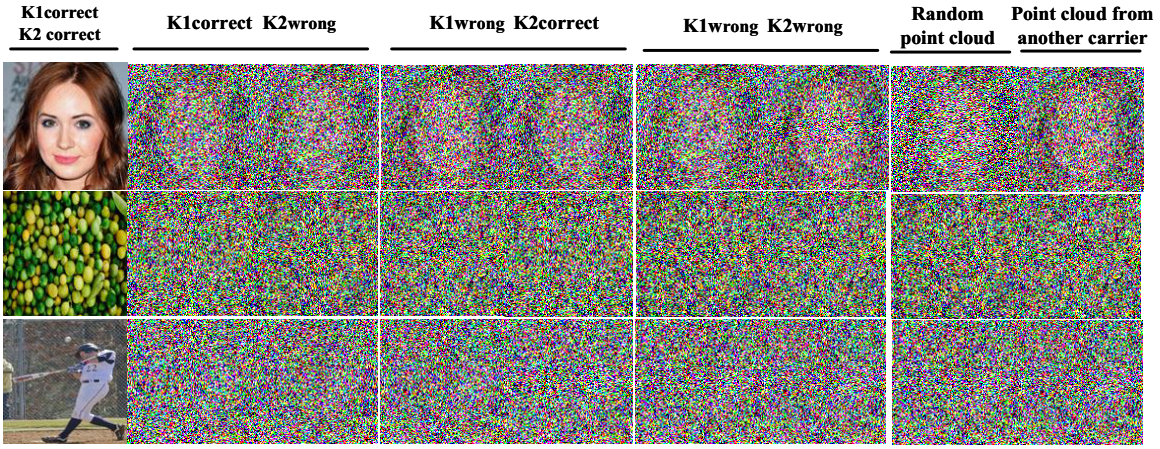

Figure 8: Secret image results under different attack settings.

Only when both the noise seed K_1 and the secret key K_2 are correct can the extractor recover a high-quality secret image. Under the correct setting, which is consistent with the DIV2K recovered-secret result in Table 2, the recovered secret image achieves a PSNR of 48.013 dB and an SSIM of 0.963. When an incorrect key, an incorrect noise seed, a random point cloud, or a point cloud from another carrier is used, the recovery quality decreases substantially, with PSNR values below 12 dB and SSIM values below 0.09. These results show that the secret recovery process is sensitive to both the noise seed and the point-selection key. Even if an attacker obtains the public carrier INR, it is difficult to construct an effective secret point cloud and recover the secret image without the correct K_1 and K_2.

4.7 Ablation Study

4.7.1 Number of Sampled Point-Cloud Points

In the above experiments, the number of sampled points is set to 1024 for carrier images with a resolution of 128×128. In this section, we fix the carrier image resolution at 128 and vary the number of sampled points to investigate the influence of the secret point-cloud size M on secret image reconstruction. As shown in Fig. 9, when the sampling number is 1024, the reconstructed secret image maintains relatively high quality while requiring a shorter extractor training time. Therefore, we use $M = 1024$ as the sampling-number hyperparameter for carrier images with a resolution of 128×128.

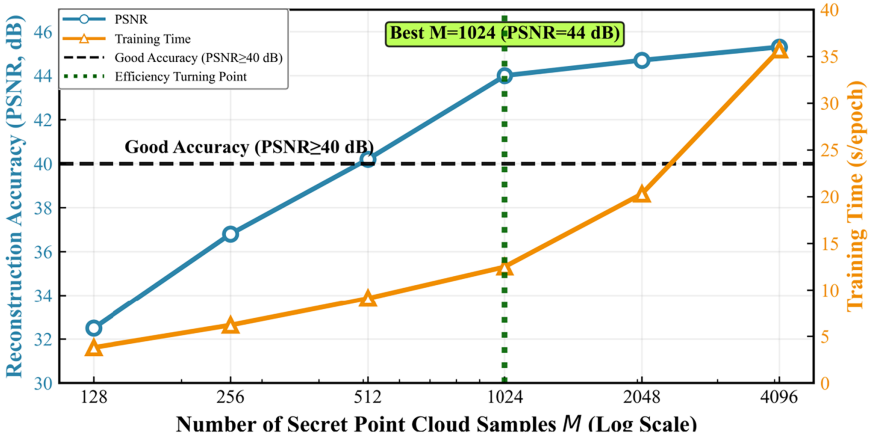

Figure 9: Relationship among the sampling number, image quality, and training time.

4.7.2 Noise Scale

In the above experiments, the noise scale is fixed at $\sigma = 10^{-4}$. The noise scale directly affects both the concealment of the secret point cloud and the reconstruction accuracy of the secret image. To analyze its influence, we conduct a gradient ablation study by changing only σ while keeping all other parameters unchanged. When the noise scale is too small, i.e., $\sigma < 5 \times 10^{-5}$, the noise is insufficient to disrupt the explicit structure of the secret point cloud, and a third party may infer hidden information from the regularity of the point cloud. As shown in Fig. 10, considering both security and recovery quality, we record the extractor training time and the recovered secret image quality under different noise scales. Based on the overall results, $\sigma = 10^{-4}$ is selected as the noise-scale hyperparameter. To visualize how noise disrupts the regular geometric structure of the point cloud, we further enlarge local high-density point-cloud surfaces of a 3D scene carrier. As shown in Fig. 11, without noise or with very small noise, the point arrangement still exhibits a highly regular grid-like structure. When stronger noise is introduced, the grid pattern is disrupted into an irregular distribution, which helps mask the physical regularity of the point cloud and improves concealment.

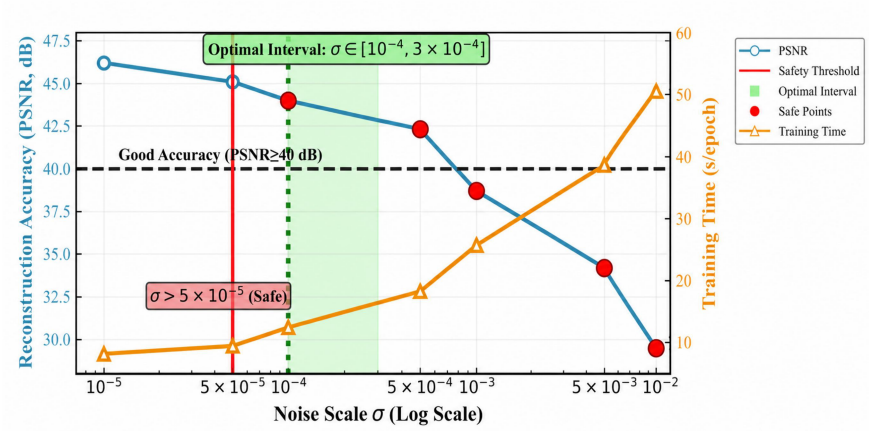

Figure 10: Effect of the noise scale on secret image quality and training time.

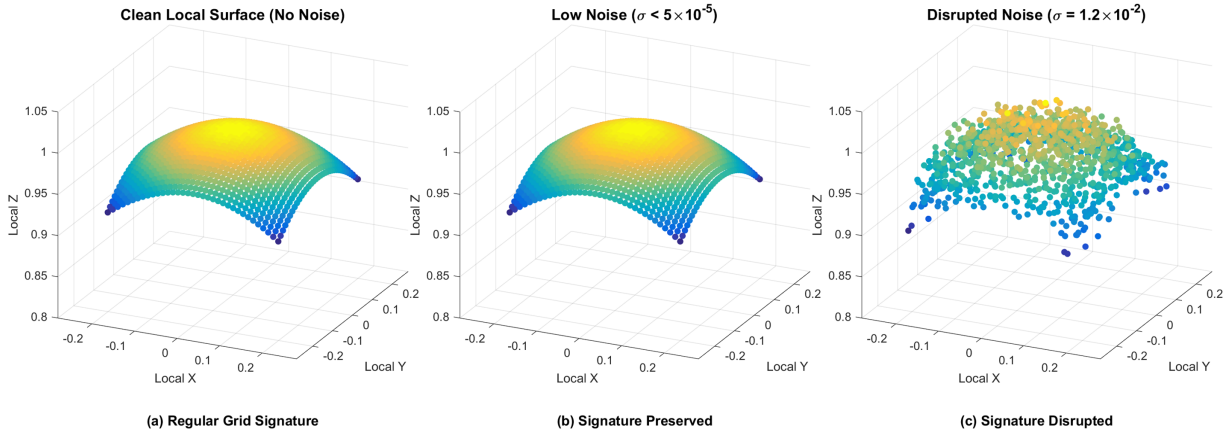

Figure 11: Visualization comparison of local 3D point-cloud surfaces under different noise scales: (**a**) noise-free local surface with a regular grid signature; (**b**) low-noise case ($\sigma < 5 \times 10^{-5}$), where the geometric grid signature remains intact; (**c**) strong-noise case ($\sigma = 1.2 \times 10^{-2}$), where the regular grid signature is disrupted.

4.8 Comparison with Steganographic Schemes

To evaluate the performance of the proposed method, we compare it with U-INR [22], INRSteg [21], and Deep Cross-Modal Steganography, denoted as Deep-CM [32]. All comparison methods are reproduced using the official code provided by the authors or carefully implemented according to the original papers when the code is unavailable. For fairness, the hyperparameters follow the default settings recommended in the corresponding papers, without additional tuning for our experiments. The training time is measured as follows: for U-INR and INRSteg, it includes both secret embedding in the parameter space and carrier fitting; for Deep-CM, it includes the complete training of the cross-modal network; for our method, it includes carrier INR fitting for 2500 epochs and extractor training for 150 epochs. All experiments are conducted on the same hardware platform, i.e., an NVIDIA GeForce RTX 2070 GPU, using the same cover–secret image pairs with a resolution of 128 × 128. For our method, the reported training time includes both carrier INR fitting and extractor training. The model size includes both the carrier INR and the extractor, where the carrier INR contains about 0.034M parameters and the extractor contains 6.358M parameters, resulting in a total model size of about 6.392M. The results of our method are reported under the DIV2K image-carrier

setting and are kept consistent with Table 2. Table 7 compares different methods in terms of cover image quality, secret image recovery quality, model size, and training time.

Table 7: Comparison with representative multimodal steganography schemes.

Method	Model Size	Cover Image		Secret Image		Training Time
		PSNR	SSIM	PSNR	SSIM	
U-INR [22]	0.25M	38.32 ± 0.84	0.989 ± 0.004	34.50 ± 1.12	0.973 ± 0.008	3.00 ± 0.12 h
INRSteg [21]	0.46M	62.31 ± 1.47	0.999 ± 0.001	43.67 ± 0.91	0.992 ± 0.004	0.55 ± 0.03 h
Deep-CM [32]	1066.24M	19.77 ± 1.25	0.519 ± 0.037	29.98 ± 1.34	0.926 ± 0.018	7.00 ± 0.26 h
Ours	6.392M	38.732 ± 1.106	0.956 ± 0.012	48.013 ± 0.697	0.963 ± 0.001	0.15 ± 0.01 h

Note: U-INR denotes Unified Steganography via Implicit Neural Representation [22]; INRSteg denotes Implicit Steganography Beyond the Constraints of Modality [21]; Deep-CM denotes Deep Cross-Modal Steganography Using Neural Representations [32]; INR denotes implicit neural representation; PSNR denotes peak signal-to-noise ratio; SSIM denotes structural similarity index measure; M denotes million trainable parameters; h denotes hours.

As shown in Table 7, the proposed method achieves competitive secret image recovery quality under the DIV2K setting, with a secret-image PSNR of 48.013 dB and an SSIM of 0.963. Compared with U-INR and Deep-CM, our method provides higher secret recovery quality and shorter training time. Compared with INRSteg, our method obtains higher secret-image PSNR and requires less training time, although its cover reconstruction quality and secret-image SSIM are not the highest. From the perspective of model structure, U-INR and INRSteg mainly hide secret information in the INR parameter space, while Deep-CM relies on a large-scale cross-modal hiding network. In contrast, our method does not modify the carrier network after carrier INR fitting, but recovers the secret image through point-cloud sampling, key-driven point selection, and an independent extractor. Overall, the proposed method shows advantages in secret image recovery quality and training efficiency, while the additional model size introduced by the extractor should also be considered in practical transmission.

4.9 Limitations

Although the proposed multimodal INR steganographic framework based on point-cloud intermediate representation shows certain advantages in multimodal adaptability, secret-image recovery quality, and training efficiency, it still has several limitations. A major limitation is that the proposed method relies on an independently trained message extractor. This design avoids directly modifying the parameters or structure of the carrier INR and decouples the carrier representation from the secret recovery process. However, the extractor remains a necessary component for reconstructing the secret information. Therefore, in practical applications, the parameter scale of the extractor, its encapsulation and transmission strategy, and its potential exposure risk should be further considered. Moreover, when the extractor is encapsulated as a submodule within a host neural network, it may alter the weight distribution of the host model, which requires further investigation. In particular, for high-resolution secret images, the output dimension and parameter number of the extractor increase with the secret-image resolution, as shown in Table 3. Therefore, future work will focus on optimizing the extractor architecture, improving the security of extractor transmission, and further validating the applicability of the proposed framework on more complex multimodal data, larger-scale experiments, stronger attack scenarios, and advanced deep-learning-based steganalysis detectors.

5 Conclusion

This paper proposes a point-cloud-based multimodal INR steganographic framework. The framework represents carriers from different modalities as carrier INRs, samples the fitted functions into reproducible noisy coordinate–feature point clouds, and constructs a key-conditioned secret point cloud using a pre-shared noise seed and secret key. A corresponding extractor is then trained to recover the secret image from this point cloud. This design further indicates that the carrier INR itself does not store or encode the secret message; instead, hiding is realized through key-conditioned point selection and the joint consistency among the carrier INR, noise seed, secret key, and extractor. In this way, image, NeRF/3D scene, and meteorological carriers are unified through a common point-cloud intermediate representation. Experiments on 128×128 image carriers show that the recovered secret images achieve PSNR/SSIM values of 47.529 dB/0.972, 46.832 dB/0.989, and 48.013 dB/0.963 on CelebA-HQ, COCO, and DIV2K, respectively. In the multimodal evaluation, the proposed method obtains secret-image PSNR values of 44.736 and 45.218 dB on NeRF-Synthetic and ERA5 carriers, respectively. Security-related experiments show that the tested steganalysis classifiers remain close to chance-level performance, with detection accuracies from 50.8% to 52.1%, and incorrect keys or inconsistent point-cloud inputs reduce the secret-image PSNR to below 12 dB. Compared with representative steganographic schemes under the DIV2K setting, the proposed method achieves a secret-image PSNR of 48.013 dB with a training time of 0.15 h. Overall, the results indicate that point-cloud intermediate representations provide a feasible interface for multimodal INR-based steganography. Future work will further investigate extractor compression, extractor-level detectability, and robustness under stronger model perturbations and advanced steganalysis settings, including deep-learning-based detectors.

Acknowledgement: The authors acknowledge the support from the General Program of the National Natural Science Foundation of China. To ensure linguistic accuracy and adherence to academic conventions, the English translation of the manuscript text and abstract was initially completed by the authors, followed by AI-assisted polishing using the DeepSeek and DouBao large language models. The authors have subsequently conducted thorough manual review, verification, and optimization of the entire content, and take full responsibility for the final accuracy and integrity of the manuscript.

Funding Statement: This research was funded by the National Natural Science Foundation of China, grant numbers 62272478, 61872384 and 62102451. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Qiya Wang and Jia Liu; methodology, Qiya Wang; software, Qiya Wang; validation, Qiya Wang, Yuwei Lu and Yujie Liu; formal analysis, Qiya Wang; investigation, Qiya Wang and Peng Luo; resources, Jia Liu; data curation, Qiya Wang; writing—original draft preparation, Qiya Wang; writing—review and editing, Jia Liu; visualization, Qiya Wang; supervision, Jia Liu; project administration, Jia Liu; funding acquisition, Jia Liu. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available within the article, and the archived version of the code supporting the findings of this study is openly available in GitHub at https://github.com/twinlj77/StegaMIR/. Additional data are available from the Corresponding Author, Jia Liu, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cheddad A, Condell J, Curran K, Mc Kevitt P. Digital image steganography: survey and analysis of current methods. Signal Process. 2010;90(3):727–52.
2. Chaumont M. Deep learning in steganography and steganalysis. In: Digital media steganography. Amsterdam, The Netherlands: Elsevier; 2020. p. 321–49.
3. Wani MA, Sultan B. Deep learning based image steganography: a review. Wiley Interdiscip Rev: Data Min Knowl Discov. 2023;13(3):e1481.
4. Hu K, Wang M, Ma X, Chen J, Wang X, Wang X. Learning-based image steganography and watermarking: a survey. Expert Syst Appl. 2024;249:123715.
5. Baluja S. Hiding images within images. IEEE Trans Pattern Anal Mach Intell. 2019;42(7):1685–97. doi:10.1109/tpami.2019.2901877.
6. Zhu J, Kaplan R, Johnson J, Li F. Hidden: hiding data with deep networks. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018 Sep 8–14; Munich, Germany. p. 657–72.
7. Jing J, Deng X, Xu M, Wang J, Hinet GZ. Deep image hiding by invertible network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021 Oct 10–17; Montreal, QC, Canada. p. 4733–42.
8. Xu Y, Mou C, Hu Y, Xie J, Zhang J. Robust invertible image steganography. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022 Jun 18–24; New Orleans, LA, USA. p. 7875–84.
9. Zhang KA, Cuesta-Infante A, Xu L, Veeramachaneni K. SteganoGAN: high capacity image steganography with GANs. arXiv:1901.03892. 2019.
10. Weng X, Li Y, Chi L, Mu Y. High-capacity convolutional video steganography with temporal residual modeling. In: Proceedings of the 2019 on International Conference on Multimedia Retrieval; 2019 Jun 10–13; Ottawa, ON, Canada. p. 87–95.
11. Kweon H, Park J, Woo S, Cho D. Deep multi-image steganography with private keys. Electronics. 2021;10(16):1906. doi:10.3390/electronics10161906.
12. Guan Z, Jing J, Deng X, Xu M, Jiang L, Zhang Z, et al. DeepMIH: deep invertible network for multiple image hiding. IEEE Trans Pattern Anal Mach Intell. 2022;45(1):372–90.
13. Luo T, Zhou Y, He Z, Jiang G, Xu H, Qi S, et al. Stegmamba: distortion-free immune-cover for multi-image steganography with state space model. IEEE Trans Circuits Syst Video Technol. 2024;35(5):4576–91.
14. Priya S, Abirami S, Arunkumar B, Mishachandar B. Super-resolution deep neural network (SRDNN) based multi-image steganography for highly secured lossless image transmission. Sci Rep. 2024;14(1):6104. doi:10.1038/s41598-024-54839-7.
15. Das A, Wahi JS, Anand M, Rana Y. Multi-image steganography using deep neural networks. arXiv:2101.00350. 2021.
16. Sitzmann V, Martel J, Bergman A, Lindell D, Wetzstein G. Implicit neural representations with periodic activation functions. Adv Neural Inf Process Syst. 2020;33:7462–73.
17. Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R. Nerf: representing scenes as neural radiance fields for view synthesis. Commun ACM. 2021;65(1):99–106. doi:10.1007/978-3-030-58452-8_24.
18. Luo P, Liu J, Ke Y, Zhang M, Mu D. Hiding functions within functions: steganography by implicit neural representations. Tsinghua Sci Technol. 2026;31(2):1058–74.
19. Dong W, Liu J, Chen L, Sun W, Pan X, Ke Y. Implicit neural representation steganography by neuron pruning. Multimed Syst. 2024;30(5):266. doi:10.21203/rs.3.rs-4417487/v1.
20. Dong W, Liu J, Chen L, Sun W, Pan X, Ke Y. StegaINR4MIH: steganography by implicit neural representation for multi-image hiding. J Electron Imaging. 2024;33(6):063017–7.
21. Song S, Yang S, Yoo CD, Kim J. Implicit steganography beyond the constraints of modality. In: European Conference on Computer Vision. Berlin/Heidelberg, Germany: Springer; 2024. p. 289–304.
22. Song Q, Luo Z, Huang X, Li S, Wan R. Unified steganography via implicit neural representation. arXiv:2505.01749. 2025.
23. Guo C, Wu R, Weinberger KQ. On hiding neural networks inside neural networks. arXiv:2002.10078. 2020.
24. Hayes J, Danezis G. Generating steganographic images via adversarial training. arXiv:1703.00371. 2017.

25. Lu SP, Wang R, Zhong T, Rosin PL. Large-capacity image steganography based on invertible neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021 Jun 3–7; Denver, CO, USA. p. 10816–25.
26. Li C, Feng BY, Fan Z, Pan P, Wang Z. StegaNeRF: embedding invisible information within neural radiance fields. In: Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision; 2023 Oct 1–6; Paris, France. p. 441–53.
27. Luo Z, Guo Q, Cheung KC, See S, Wan R. CopyRNeRF: protecting the copyright of neural radiance fields. In: Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision; 2023 Oct 2–6; Paris, France. p. 22401–11.
28. Xu Z, Xu D, Li Z, Hu J, Zheng B, Zhang C, et al. StegaFusion: steganography for information hiding and fusion in multimodality. Inf Fusion. 2026;131:104150.
29. Jiang J, Wang Z, Yuan Z, Zhang X. Generative image steganography based on text-to-image multimodal generative model. IEEE Trans Circuits Syst Video Technol. 2025;35(9):8907–16. doi:10.1109/tcsvt.2025.3556892.
30. Chang CC, Echizen I. Steganography beyond space-time with chain of multimodal AI. Sci Rep. 2025;15(1):12908. doi:10.1038/s41598-025-97238-2.
31. Kishore DR, Suneetha D, Babu PN, Chinababu P. Deep convolutional neural network-based image steganography technique for audio-image hiding algorithm. IJEAT. 2020;9(4):2187–9. doi:10.35940/ijeat.d7843.049420.
32. Han G, Lee DJ, Hur J, Choi J, Kim J. Deep cross-modal steganography using neural representations. In: Proceedings of the 2023 IEEE International Conference on Image Processing (ICIP); 2023 Oct 8–11; Kuala Lumpur, Malaysia. p. 1205–9.
33. Karras T, Aila T, Laine S, Lehtinen J. Progressive growing of GANs for improved quality, stability, and variation. arXiv:1710.10196. 2017.
34. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: common objects in context. In: Computer Vision—ECCV 2014 (ECCV 2014). Berlin/Heidelberg, Germany: Springer; 2014. p. 740–55.
35. Timofte R, Agustsson E, Van Gool L, Yang MH, Zhang L. Ntire 2017 challenge on single image super-resolution: methods and results. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops; 2017 Jul 21–26; Honolulu, HI, USA. p. 114–25.
36. Hersbach H, Bell B, Berrisford P, Biavati G, Horányi A, Muñoz Sabater J, et al. ERA5 monthly averaged data on single levels from 1979 to present. Copernic Clim Change Serv Clim Data Store. 2019;10:252–66.
37. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process. 2004;13(4):600–12.
38. Ker AD. Steganalysis of LSB matching in grayscale images. IEEE Signal Process Lett. 2005;12(6):441–4. doi:10.1109/lsp.2005.847889.