East Journal of Computer Science
ISSN: 3079-9406

Volume1, Issue 4
Publisher: East Publication & Technology
DOI: https://doi.org/10.63496/ejcs.Vol1.Iss4.188

StyleGAN2-Stego: Secure Coverless Image Steganography via Latent Space Encoding

Ari Ibrahim Hamid*[1], Wafaa Mustafa Abduallah[2]

[1]Department of Information Technology Management, Technical College of Administration, Duhok Polytechnic University, Kurdistan Region, Iraq, ari.hamid@dpu.edu.krd.
[2]Department of Cyber Security Engineering, Technical College of Engineering, Duhok Polytechnic University, Kurdistan Region, Iraq, wafaa.abduallah@dpu.edu.krd.
*Corresponding Author.

Received: 20/08/2025, **Revised**: 21/08/2025, **Accepted**: 29/08/2025, **Published**: 30/08/2025

***Abstract*:**

Coverless image steganography (CIS) enhances secrecy by avoiding direct modifications to existing images, unlike traditional methods that embed data by altering image content. However, many existing approaches struggle to balance payload capacity, visual quality, and resistance to steganalysis. This paper introduces StyleGAN2-Stego, a generative steganography framework that conceals secret messages in the latent space of a pre-trained StyleGAN2-ADA generator. The framework consists of an encoder that transforms a binary message into a latent perturbation vector, a fixed generator that synthesizes realistic stego images from the modified latent space, and a decoder that reconstructs the original message from the generated image. The encoder and decoder are trained jointly, while the generator remains frozen to preserve visual fidelity. Experimental results show that StyleGAN2-Stego achieves a payload of 0.5 bits per pixel, a message recovery accuracy of 98.72%, and strong resistance to detection by advanced steganalysis tools such as Xu-Net and Ye-Net. It also produces high-quality images, with a Fréchet Inception Distance (FID) of 6.78. However, further evaluation under real-world image distortions such as compression, resizing, and scaling is required to validate robustness in practical communication scenarios. These findings highlight the potential of StyleGAN2-Stego for real-world applications such as secure digital communication, copyright protection, and digital watermarking.

***Keywords*: Coverless Image Steganography, Steganalysis, Generative Steganography, Deep Learning, StyleGAN2-ADA, Latent Space Encoding, Secure Communication.**

1. Introduction

The rapid expansion of digital communication has transformed how information is created, shared, and stored. Alongside these advances, the need for protecting sensitive data has become more critical than ever. Steganography, the practice of concealing secret information within ordinary media, has emerged as an important technique for enabling covert communication without arousing suspicion [1], [2]. Unlike cryptography [3], which protects the content of a message but leaves its existence visible, steganography hides the very presence of the message, making detection substantially more difficult. It can be adapted to various forms of digital media, encompassing images [4-8], videos [9], audio [10], and textual information [11]. Digital images are widely used in steganography. They are common in daily communication and can carry large amounts of hidden data. The image that contains the embedded secret data is referred to as a stego image. Techniques such as steganalysis [12-14] are employed to detect the presence of hidden information in such images.

Traditional image steganography methods often rely on modifying certain components of an existing image, such as the least significant bits (LSBs) of pixel values [15] or texture-rich areas [16]. While these approaches can embed large amounts of information, they inevitably introduce subtle changes to the cover image that may be detectable by modern steganalysis tools. In particular, deep learning–based steganalyzers such as Xu-Net [17] and Ye-Net [18] have demonstrated a high ability to detect even minor alterations, reducing the security of conventional embedding techniques.

To overcome these limitations, coverless image steganography (CIS), and in particular generative steganography (GS), has gained significant attention. Rather than embedding secret data into a pre-existing cover, generative methods directly generate the stego image from the secret message using generative models [19]. Since there is Early generative steganography approaches, such as those based on DCGANs [20] or WGAN-GP [21], demonstrated the potential of using deep generative models to produce stego images without modifying an existing cover. However, despite these promising directions, they exhibited several critical limitations. In particular, DCGAN-based methods often suffered from unstable training and mode collapse, which restricted their ability to generate diverse and consistent images. WGAN-GP introduced improvements in training stability, but the resulting images were typically limited to low resolutions, which constrained their practical applicability. Moreover, both approaches struggled to achieve reliable message recovery at higher payloads, reducing their usefulness for robust communication. These limitations highlighted the need for more stable and controllable generative models.

More recent studies incorporated attention mechanisms [22] and disentanglement strategies [23] to improve image quality and robustness, while methods such as SSteGAN [24] and sampling-based frameworks [25] advanced flexibility in embedding. Building on this progression, researchers explored adversarial training for steganography without embedding [26], message-conditioned sampling [28], and decoupled generator–decoder designs [29]. Most recently, diffusion-based methods [30] achieved state-of-the-art image quality but introduced higher computational costs and limited stability at increased payloads.

In this work, we introduce StyleGAN2-Stego, a StyleGAN2-ADA-based coverless image steganography framework that embeds information into the latent space of a pre-trained generator. Instead of altering image pixels, the proposed method modifies the generator's internal latent representation by adding a small, learned adjustment derived from the secret message. To the best of our knowledge, this is the first framework that leverages a fixed StyleGAN2-ADA generator for latent-space steganography, ensuring stable training and improved fidelity compared to retrainable diffusion or GAN-based approaches.

We evaluate StyleGAN2-Stego with a payload capacity of 0.5 bits per pixel (bpp) and assess its robustness using two widely used steganalysis networks, Xu-Net and Ye-Net. The framework achieves strong undetectability, with detection error rates close to random guessing, and produces visually realistic images, confirmed by a Fréchet Inception Distance (FID) of 6.78. These results highlight its potential as an effective solution for secure image-based communication using deep generative models.

The rest of this paper is organized as follows: Section 2 reviews related work in generative and coverless image steganography. Section 3 describes the architecture and methodology of StyleGAN2-Stego. Section 4 presents the experimental setup, results, and analysis. Finally, Section 5 concludes the study and discusses potential future work.

2. Related Work

Early steganography methods primarily relied on directly modifying existing cover images, such as altering pixel values or embedding data in specific image regions. While these approaches were effective to some extent, they are increasingly vulnerable to modern detection techniques. To address this limitation, generative steganography (GS) has emerged as a compelling alternative. Rather than embedding messages into pre-existing images, GS techniques generate new images from scratch with the secret message inherently encoded. Since there is no original cover image for comparison, detecting hidden data becomes substantially more challenging.

One of the foundational works in this space was presented by Hu et al. [20] in 2018, who employed a Deep Convolutional Generative Adversarial Network (DCGAN) to produce stego images. In their system, the secret message was embedded into the noise vector fed to the generator. This innovation removed the need for a separate cover image and introduced the concept of message-conditioned image generation. However, the generated visuals were often of low quality, and decoding reliability varied significantly based on input complexity. Furthermore, DCGANs at the time were prone to unstable training and lacked fine-grained control, limiting real-world applicability.

In the same year, Wang et al. [24] (2018) introduced SSteGAN, a self-learning generative steganography model. SSteGAN took both noise and secret data as inputs and trained the generator and decoder jointly without manual supervision. The model treated steganography as a learning task, allowing it to discover how to hide and recover information automatically. It achieved a balance between visual quality and decoding reliability, although training stability and parameter tuning added complexity.

Building on these foundations, Li et al. [21] in 2020 proposed a system based on Wasserstein GANs with Gradient Penalty (WGAN-GP). This model improved convergence and addressed mode collapse, producing more consistent results. The use of the Wasserstein loss function, coupled with gradient penalties, facilitated smoother training dynamics. While image quality improved compared to DCGAN-based methods, the model still required careful embedding strategies, and decoding accuracy remained sensitive to input variations and noise.

A different line of advancement was proposed by Yu et al. [22] in 2021, who introduced SAGAN-Steg, a steganography framework that avoided traditional embedding by leveraging attention-enhanced GANs. Their architecture included a generator that synthesized images conditioned on both noise and secret data, and an extractor that was responsible for retrieving the embedded message. To improve the model's focus on meaningful latent features, they incorporated attention mechanisms into both components. The authors also introduced a soft-margin adversarial loss, which enhanced robustness against noise and led to more stable training. The resulting system produced more realistic images and achieved high message extraction accuracy, even under slight distortions.

A notable contribution came from Liu et al. [23] in 2022, who proposed IDEAS, an Image Disentanglement Autoencoder for Steganography designed to hide information without explicit embedding. Rather than injecting secret data directly into image pixels or noise vectors, their method disentangled the generative process into two separate latent spaces: one responsible for visual content and the other for the encoded message. This design enabled the model to implicitly embed information into high-level image features, improving imperceptibility and reducing visible artifacts. However, maintaining a strict separation between message and content representation posed training challenges, particularly in preserving image quality during message reconstruction.

Zhang et al. [28] in 2024 introduced a message-conditioned generative steganography framework that eliminates the need for traditional embedding. In their approach, the secret message is mapped into a continuous latent noise vector through a defined mapping function with a carrier component. This vector is then passed to the generator, which synthesizes a realistic image that inherently carries the hidden information. On the receiving side, extractor networks recover a secret tensor from the generated image, and an inverse mapping reconstructs the original message. By embedding the payload at the earliest stage of generation, this design avoided pixel-level modifications and improved imperceptibility while maintaining flexibility in message recovery.

In the same year, Ren and Wu [29] (2024) introduced JoCS, a robust joint coverless image steganography framework based on two independent modules: a generator and a decoder. Unlike prior models that train these components together, JoCS kept them separate. The generator created visually convincing stego images conditioned on the secret message, while the decoder independently learned to recover the embedded data. This decoupled design offered greater flexibility, as the decoder could be retrained separately for different message types without modifying the generator. A feedback mechanism was also incorporated to adjust generation based on recovery performance. Although this improved robustness against noise and data loss, the lack of joint training sometimes resulted in weaker coordination between image realism and message recovery.

Most recently, Kim et al. [30] in 2025 proposed Diffusion-Stego, a training-free steganography framework that leveraged the powerful generative capabilities of diffusion models. Instead of training a custom model, they utilized a pre-trained diffusion model and introduced a method called message projection. This technique projected the secret message onto the latent noise space of the diffusion model, meaning the message was encoded by slightly altering the random noise input used to generate the image. These subtle modifications guided the diffusion sampling process so that the final image inherently contained the hidden information. Since no pixel-level modification occurred, the resulting stego images were visually indistinguishable from naturally generated samples. This approach was highly efficient, as it avoided retraining and fully exploited the image quality achievable with diffusion models. However, its reliability depended heavily on the precision of the projection

mechanism, and decoding accuracy could drop when images were subjected to compression, resizing, or other distortions.

Collectively, these studies highlight the field's ongoing shift toward more secure, high-capacity, and imperceptible steganographic techniques. The progression of research from DCGAN-based methods in 2018 to diffusion-based approaches in 2025 has paved the way for increasingly advanced and secure systems. Building on these prior contributions, our proposed framework, StyleGAN2-Stego, introduces a refined approach that operates at the latent level of a pre-trained StyleGAN2-ADA generator. Instead of altering image pixels directly, it embeds secret messages by subtly modifying the generator's internal representation before image synthesis. This controlled adjustment enables the model to produce natural-looking images that conceal data effectively. By leveraging the capabilities of modern generative models, the method achieves high data embedding rates, strong imperceptibility, and robust resilience against deep learning-based steganalysis techniques.

3. The Proposed Framework

3.1 Overview of the Proposed Framework

The StyleGAN2-Stego framework introduces a coverless image steganography technique that embeds secret messages by applying small, calculated changes to the latent input of a generative model. Specifically, it adds a latent perturbation vector Δw derived from the secret message m to the latent vector w that is passed into a pre-trained StyleGAN2-ADA generator G. This subtle modification allows the model to produce photorealistic stego images I_s that visually resemble normal outputs while securely encoding the hidden data in a way that is difficult for modern steganalysis tools to detect. The proposed framework is built around three key components, each playing a distinct role in the message embedding and extraction process:

1. **Encoder E**: Transforms the binary representation of a secret message b into a latent perturbation vector. This vector subtly adjusts the generator's latent space to encode the message without altering visible image content.
2. **Generator G**: Utilizes a fixed, pre-trained StyleGAN2-ADA model that synthesizes the final stego image. It receives a modified latent vector w', which is the result of adding the perturbation vector to the original latent input.
3. **Decoder D**: Processes the generated stego image to accurately recover the embedded message from the high-level features introduced during generation.

This separation of responsibilities ensures that the generator retains its image quality, while the encoder E and decoder D learn to perform accurate and robust message embedding and extraction. Figure 1 illustrates the message encoding, image generation, and decoding process.

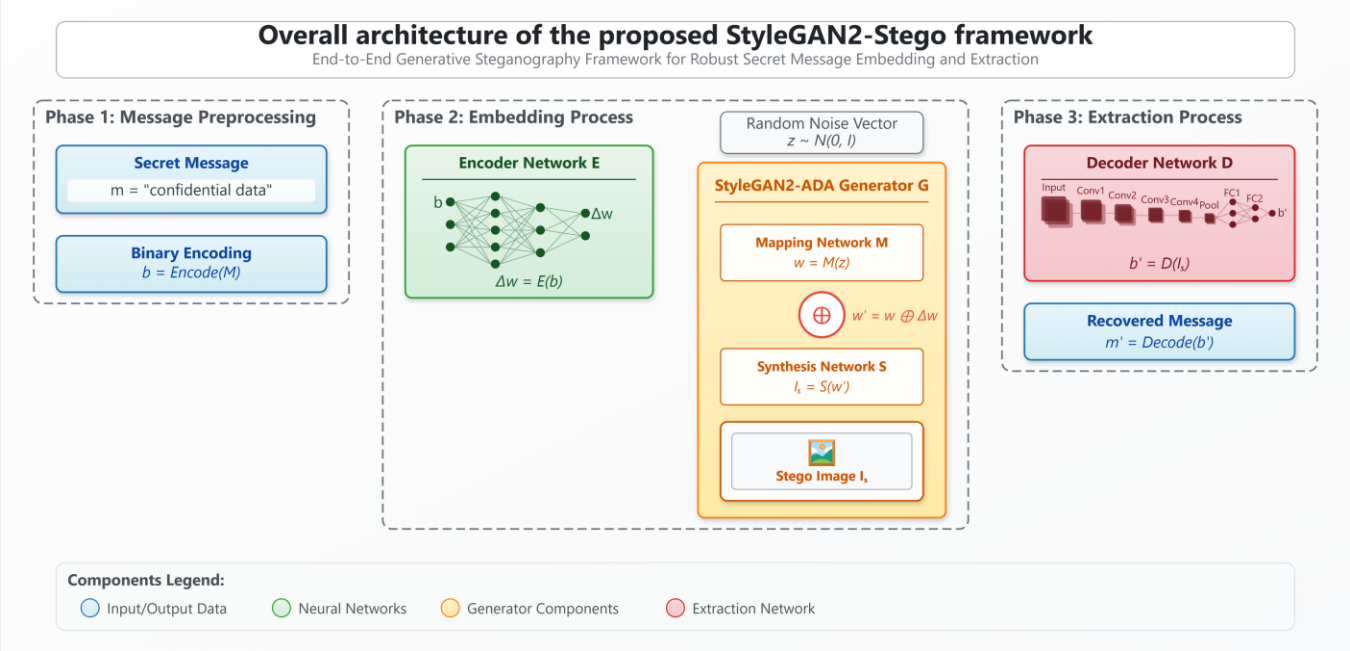

Figure 1: Overall architecture of the StyleGAN2-Stego framework. The encoder maps the binary message into a latent perturbation vector, which is added to the StyleGAN2-ADA latent representation. The generator synthesizes a stego image, and the decoder reconstructs the embedded message.

3.2 Message and Image Preprocessing

The system processes two input types: a secret message m and the corresponding generated stego image I_s.

3.2.1 Message Preprocessing

To ensure compatibility with the encoder E, the secret message first goes through a preprocessing phase. The message is either truncated or padded with null characters to reach a fixed length of 4096 characters. This guarantees a uniform input size regardless of the original message length. The standardized message is then translated into a binary representation using a text-to-bits function, where each character becomes an 8-bit binary representation. The resulting binary array is expressed as:

$$b = Encode(m), \quad b \in \{0,1\}^n \tag{1}$$

where n represents the total number of bits in the encoded secret message. The resulting bit sequence is used as input for the encoder, ensuring both consistent formatting and the bit-level precision needed for accurate encoding.

3.2.2 Image Preprocessing

The StyleGAN2-ADA generator G creates images with pixel values ranging between –1 and 1. Before being processed by the decoder D, these images are normalized to the [0, 1] range to maintain compatibility. To avoid artifacts from lossy compression, the images are saved in PNG format. This normalization step ensures the decoder always receives inputs in a consistent format, improving the reliability of both training and message extraction.

3.3 Encoder Network

The encoder E is a fully connected neural network that transforms the binary secret message b into a latent perturbation vector Δw. This vector alters the internal latent representation w used by the StyleGAN2-ADA generator G. As illustrated in Figure 2, the encoder architecture is composed of three fully connected layers, which are followed by a reshaping operation to align the output with the StyleGAN2-ADA input format.

- **Input Layer**: Accepts the binary message b as a one-dimensional input vector representing the full bit sequence of the secret message.
- **Hidden Layer 1**: Projects the input to a 2048-dimensional feature space, with ReLU activation to help the network capture high-level patterns in the message.
- **Hidden Layer 2**: Expands the feature representation to 4096 dimensions, increasing network's capacity to encode more detailed information.
- **Output Layer**: Maps the high-dimensional features from the previous hidden layer into the features of a flat vector of size $N_s \times D_l$. No activation function is applied at this stage.
- **Reshape Operation**: Reshapes the output into a tensor of shape [*batch*, N_s, D_l], where *batch* = -1 denotes the dynamic batch size. This shape aligns with the style input format expected by the StyleGAN2-ADA generator.

The resulting latent perturbation vector can be expressed as:

$$\Delta w = E(b), \quad \Delta w \in R^{N_s \times D_l} \tag{2}$$

where $N_s = 18$ is the number of style inputs and $D_l = 512$ is the latent dimensionality in StyleGAN2-ADA. This Δw is then added element-wise to the base latent vector w, which is generated by passing a random noise vector z through the StyleGAN2-ADA mapping network M. This combination allows the secret message to be embedded by subtly modifying the generator's internal latent space, rather than altering image pixels directly.

Figure 2: Encoder network architecture for StyleGAN2-Stego, mapping the binary secret message into a perturbation vector Δw aligned with the StyleGAN2-ADA latent space

3.4 StyleGAN2-ADA Generator

StyleGAN2-Stego employs a StyleGAN2-ADA generator G pre-trained on the FFHQ dataset, which is widely used for generating high-quality human face images. The generator's mapping network M transforms a random noise vector z into an intermediate latent vector w. This vector is then modified by adding the latent perturbation Δw, produced by the encoder E, resulting in a modified latent vector w'. This modified vector is passed to the synthesis network S, which produce the final high-resolution stego image I_s. The process can be formally expressed as:

$$I_s = S(w'), \quad w' = w + \Delta w, \quad w = M(z) \tag{3}$$

where $M(\cdot)$ is the mapping network that generates the base latent vector w from the noise input z, and Δw is the perturbation vector provided by the encoder. During training, all generator parameters are kept frozen. This design choice ensures that image quality is preserved and that the generator's output distribution remains stable. As a result, any variations in the generated images are entirely due to the encoder's controlled adjustments to the latent representation, not from changes in the generator itself.

It is important to note that the StyleGAN2-ADA generator used in this work was pre-trained on the FFHQ dataset, which primarily contains human face images. Consequently, the results are domain-specific to facial imagery. Extending the framework to more diverse datasets, such as natural scenes or objects, is an important direction for future work to validate generalization across broader visual domains.

3.5 Decoder Network

The decoder D is a convolutional neural network (CNN) that recovers the embedded binary message b' from the generated stego image I_s. Its architecture is optimized to detect subtle changes introduced in the latent space of the generator, without relying on pixel-level modifications. As illustrated in Figure 3, the decoder D consists of the following components:

- **Convolutional Layers**: Four convolutional layers process the input image (3 channels). Each layer uses a 4×4 kernel, stride of 2, and padding of 1. The number of channels increases progressively from 64 to 512. Batch normalization is applied after the second, third, and fourth layers, followed by ReLU activation to ensure stable learning and non-linearity.
- **Adaptive Average Pooling**: The feature map is downsampled to a fixed size of 4×4, enabling the network to handle high-resolution images while maintaining consistent output dimensions.
- **Flattening and Fully Connected Layers**: The pooled feature map is flattened and passed through two fully connected layers. The first projects the data from 8192 (i.e., 512×4×4) to 1024 dimensions, followed by ReLU activation. The second maps it to the target message length n, corresponding to the total number of bits in the original secret message, producing a vector of bitwise probabilities.
- **Output Layer**: A Sigmoid activation is applied at the output to constrain values between 0 and 1. These values are thresholded to reconstruct the original binary message.

The decoding process can be formally expressed as:

$$b' = D(I_s), \quad b' \in \{0,1\}^n \tag{4}$$

where $D(\cdot)$ is the decoder network, I_s is the generated stego image, and b' is the recovered binary message of length n. During training, the decoder learns to identify and extract the hidden message from subtle features embedded in the generated image. These features are not directly visible but result from controlled perturbations made to the generator's internal latent representation. By training the decoder jointly with the encoder, the system learns to reliably recover the original message while ignoring unrelated variations in the image content.

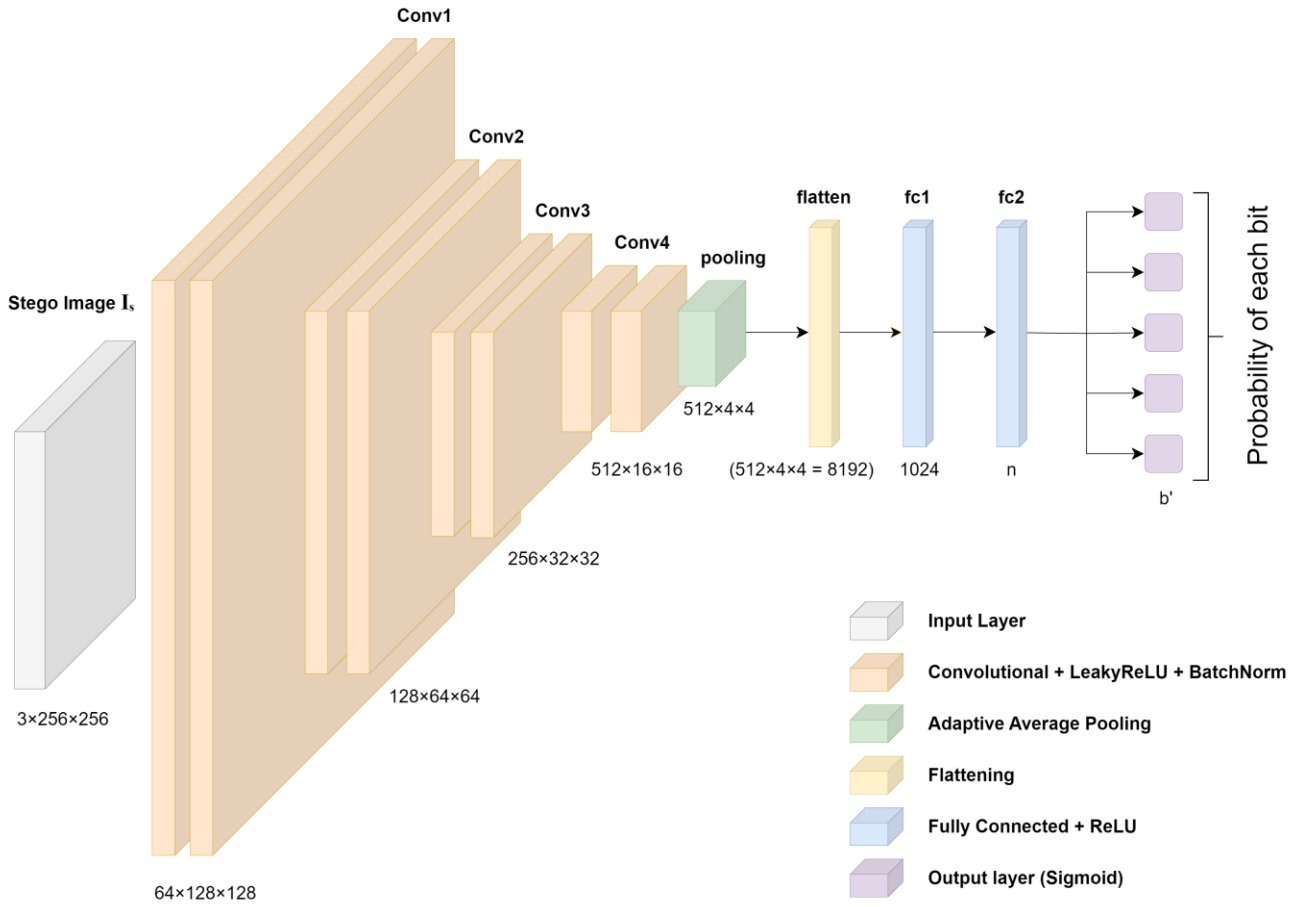

Figure 3: Decoder network architecture, designed to recover the embedded binary message from generated stego images by identifying subtle latent-space features.

3.6 Loss Function

The training objective of StyleGAN2-Stego is to simultaneously maximize message reconstruction accuracy while preserving the visual fidelity of the generated images. This is accomplished by minimizing a weighted combination of two loss functions: secret loss and latent consistency loss.

3.6.1 Secret Loss

The secret loss measures how accurately the decoder D recovers the hidden message from the generated stego image I_s. It is computed using binary cross-entropy (BCE) between the predicted binary message vector b' and the original binary message b. The loss is defined as:

$$\mathcal{L}_{secret} = \text{BCE}(b', b) \tag{5}$$

This loss encourages the decoder to produce bitwise outputs that closely match the original message, thereby ensuring high message recovery accuracy.

3.6.2 Latent Consistency Loss

To maintain the photorealism of generated stego images, the encoder's E output perturbation vector Δw is scaled and added to the base latent vector w, which is produced by the StyleGAN2-ADA mapping network M. To prevent excessive distortion in the latent space, a regularization term is introduced:

$$\mathcal{L}_{latent} = ||\Delta w||_2^2 \tag{6}$$

This latent consistency loss penalizes large deviations in the latent representation, ensuring that image realism is preserved while still allowing meaningful message encoding.

3.6.3 Total Loss

The overall training loss combines both objectives, weighted by a balancing hyperparameter λ:

$$\mathcal{L}_{total} = \mathcal{L}_{secret} + \lambda \cdot \mathcal{L}_{latent} \tag{7}$$

In our experiments, we set $\lambda = 0.5$, based on empirical tuning to achieve an optimal trade-off between message recovery and image quality. Both the encoder E and decoder D are jointly trained to minimize this total loss, ensuring robust performance in both embedding and extraction tasks.

3.7 Training Procedure

The encoder E and decoder D networks are trained jointly, while the generator (G) remains fixed to preserve the quality of synthesized stego images I_s. Training is conducted using the Adam optimizer, with parameters selected for stable convergence. The training procedure involves the following steps:

1. A random noise vector z is sampled from a standard normal distribution and mapped to a latent vector using the generator's mapping network M.
2. The binary secret message is encoded into a latent perturbation vector by the encoder.
3. The perturbation vector is added to the base latent vector to obtain the modified latent representation w', which is then passed through the generator to synthesize the stego image.
4. The generated stego image is passed to the decoder, which attempts to reconstruct the original binary message.
5. The total loss is computed by combining the secret loss and latent consistency loss. Based on this loss, the encoder and decoder parameters are updated.

On average, encoding and decoding a single 256×256 stego image required approximately 10–20 ms for embedding and about 2–5 ms for extraction on an NVIDIA RTX 4070 Ti Super GPU. demonstrating the model's suitability for near real-time applications. For full training configuration details, including learning rate, number of epochs, and image resolution, refer to Section 4.1.

4. Experimental Results and Discussion

4.1 Experimental Setup

The proposed StyleGAN2-Stego framework was implemented using a pretrained StyleGAN2-ADA generator G, originally trained on the FFHQ dataset, as the backbone for coverless stego image synthesis. All experiments were conducted in Python using the PyTorch deep learning framework on a workstation equipped with an NVIDIA GeForce RTX 4070 Ti Super GPU (16 GB VRAM) and 32 GB RAM. During training, the generator's weights were kept frozen to preserve its learned image distribution. The generation process began by sampling a 512-dimensional noise vector $z \sim \mathcal{N}(0, I)$. Which was mapped through the generator's mapping network M into a latent representation w. The encoder then produced a perturbation vector Δw from the binary secret message, which was added to w before passing through the generator's synthesis network S to produce the stego image I_s. The encoder–decoder network was trained for 120 epochs using the Adam optimizer with a learning rate of 1×10^{-5}. All generated stego images had a fixed resolution of 256×256 pixels and were stored in PNG format to avoid compression artifacts.

The performance of the proposed method was evaluated using four metrics: Bits per pixel (bpp) measured the payload capacity of stego images:

$$bpp = \frac{len(d)}{W \times H} \tag{8}$$

where $len(d)$ is the length of the embedded bit sequence, W and H are the image width and height. Extraction accuracy (Acc) quantified the proportion of correctly recovered bits:

$$\text{Acc} = \frac{d \odot d'}{\text{len}(d)} \tag{9}$$

Here, d and d' represent the original and recovered bit sequences, and $\odot$ denotes the element-wise XNOR operation. Detection error probability (P_e) was used to assess steganalysis resistance:

$$P_e = min\, P_{FA} \frac{1}{2}(P_{FA} + P_{MD}) \tag{10}$$

where, P_{FA} and P_{MD} are the false alarm rate and missed detection rate. P_e ranges in [0, 1], with a value of $P_e = 0.5$ indicating that steganalysis tool cannot distinguish between stego and non-stego images. Lastly, the Fréchet Inception Distance (FID) [27] assessed perceptual image quality by comparing the feature distributions of generated stego images with those of real images.

For benchmarking, we compared StyleGAN2-Stego with five representative coverless steganography methods: SAGAN-Steg [22], IDEAS [23], Zhang [28], JoCS [29], and Diffusion-Stego [30].

4.2 Payload Capacity

As shown in Table 1, StyleGAN2-Stego achieves an embedding capacity of 32,768 bits per image, corresponding to 0.5 bpp for 256 × 256 images. This outperforms SAGAN-Steg, IDEAS, Zhang and JoCS in capacity while maintaining high visual fidelity. Although Diffusion-Stego supports higher theoretical payloads, its performance deteriorates with increasing capacity due to degradation in visual quality and security. Note that Diffusion-Stego reports a wide range of payloads depending on sampling configurations, whereas values for other methods are as reported in their original works.

Table 1: Payload Capacities of Compared Methods

Methods	Capacity (bits/image)	Image Size	Payload (bpp)
SAGAN-Steg [22]	1200	64 × 64	0.293
IDEAS [23]	1536	256 × 256	0.0234
Zhang [28]	2304	256 × 256	0.0352
JoCS [29]	48	256 × 256	0.00073
Diffusion-Stego [30]	4096 – 24576	64 × 64	1 – 6
StyleGAN2-Stego	**32768**	**256 × 256**	**0.5**

4.3 Training Convergence and Message Recovery

The training process of StyleGAN2-Stego exhibited stable convergence throughout the 120 epochs. As shown in Figure 4, extraction accuracy increased steadily from the random initialization baseline to a final value of 98.72%. This gradual, consistent improvement indicates that the encoder–decoder network adapted effectively to the embedding and extraction tasks without overfitting or instability.

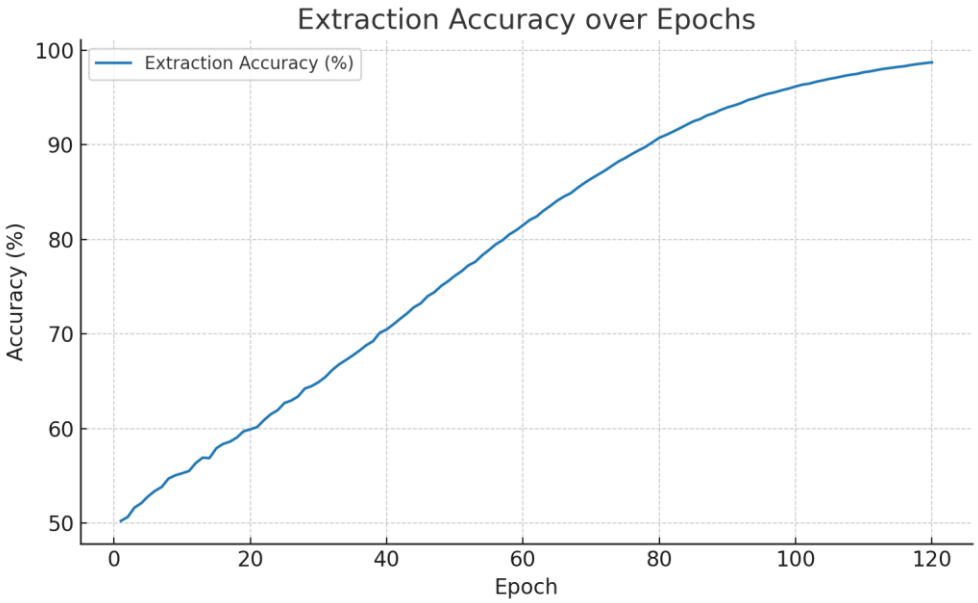

Figure 4: Extraction accuracy over 120 training epochs.

The loss curves in Figure 5 provide further insight into this convergence behavior. The secret loss declined smoothly over the course of training, reflecting the network's growing ability to reconstruct hidden messages with high fidelity. In parallel, the Δw loss stayed close to zero for most of the training process, indicating that the latent space perturbations were minimal and did not compromise the visual quality of the generated stego images.

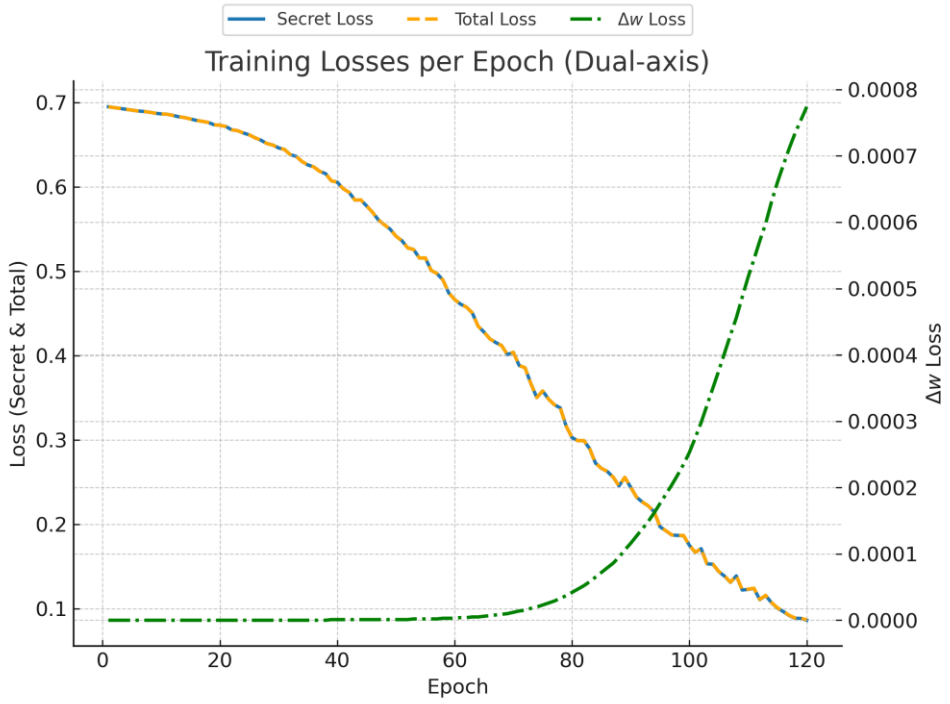

Figure 5: Training losses per epoch (secret loss and Δw loss)

Under the final experimental settings, the model achieved a message recovery accuracy of 98.72% for a payload of 0.5 bpp. This result confirms the framework's ability to reliably retrieve embedded messages while maintaining imperceptibility and robustness against detection.

4.4 Security Assessment Using Steganalysis

An essential requirement for any steganographic system is resilience against detection by steganalysis tools. In this study, the security of StyleGAN2-Stego was evaluated using two well-known CNN-based steganalyzers: Xu-Net and Ye-Net. These models were trained on 5,000 stego and 5,000 non-stego images, with an additional 1,000 images per class reserved for testing. StegoGAN-ADA was employed to generate stego images, while non-stego images were synthesized by the original pre-trained StyleGAN2-ADA generator. The steganalyzers output a classification decision indicating whether an image is suspected of containing hidden information. The detection performance is expressed using the detection error probability (P_e) defined in Section 4.1. A P_e value close to 0.5 indicates that the steganalyzer performs no better than random guessing. Some baseline methods did not report steganalysis performance in their original publications (indicated by '–').

Table 2: Steganalysis Resistance (Pe) Using Xu-Net and Ye-Net

Methods	**Xu-Net**	**Ye-Net**
SAGAN-Steg [22]	–	–
IDEAS [23]	0.403	0.535
Zhang [28]	0.497	0.488
JoCS [29]	**0.5025**	**0.5**
Diffusion-Stego [30]	0.427 - 0.385	–
StyleGAN2-Stego	**0.4860**	**0.4940**

StyleGAN2-Stego's results are very close to the optimal P_e of 0.5, indicating that it is extremely difficult for steganalyzers to distinguish its stego images from non-stego images. Figure 6 presents the Receiver Operating Characteristic (ROC) curves for Xu-Net and Ye-Net applied to StyleGAN2-Stego stego images. Both curves remain close to the diagonal, confirming that these steganalyzers perform only slightly better than random guessing. This visual evidence reinforces the near-optimal P_e values reported in Table 2.

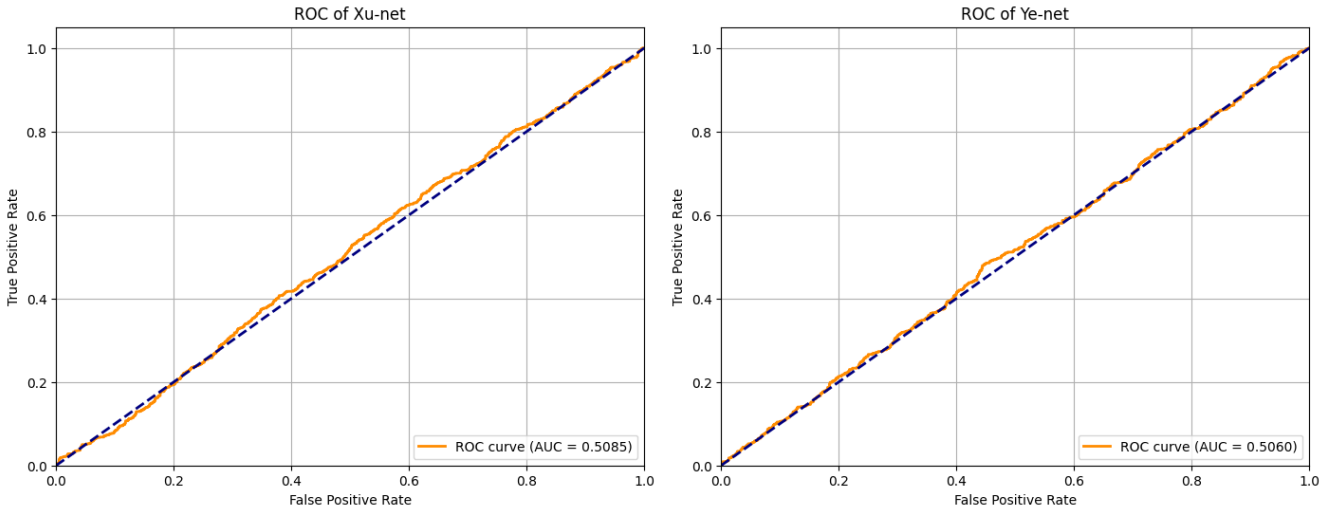

Figure 6: Roc Curves of Xu-net and Ye-net models

4.5 Security Assessment Using Image Quality

In addition to resisting steganalysis, a secure steganographic method must ensure that its stego images are visually indistinguishable from non-stego images. This prevents human observers or automated visual inspection from suspecting the presence of hidden information. To assess the visual quality of the generated stego images, we conducted both quantitative and qualitative assessments. For the quantitative evaluation, the Fréchet Inception Distance (FID) was employed, as shown in Table 3. The FID score measures the distribution difference between the generated images and real images in a deep feature space, with lower values indicating higher similarity. To evaluate our method, we computed the FID between 10,000 real FFHQ images and 10,000 stego images generated

by StyleGAN2-Stego. StyleGAN2-Stego achieved an FID value of 6.7762, which demonstrates that the generated stego images are visually similar to real images while maintaining a relatively high payload of 0.5 bpp.

Compared with SAGAN-Steg, IDEAS, Zhang, and JoCS, StyleGAN2-Stego produces superior visual quality. Although Diffusion-Stego achieves a slightly lower FID, its visual quality advantage is marginal and comes at the expense of reduced robustness and security at higher embedding capacities. Overall, StyleGAN2-Stego offers the best balance of imperceptibility, robustness, and practicality.

Table 3: Visual Quality (FID) Comparison

Methods	**FID**
SAGAN-Steg [22]	30.81
IDEAS [23]	26.37
Zhang [28]	–
JoCS [29]	10.16
Diffusion-Stego [30]	**2.77 – 3.37**
StyleGAN2-Stego	**6.7762**

In addition to quantitative evaluation, qualitative analysis was performed. Figure 7 presents visual comparisons of stego images generated by StyleGAN2-Stego and other state-of-the-art methods, including SAGAN-Steg [22], IDEAS [23], JoCS [29], and Diffusion-Stego [30]. These results confirm that StyleGAN2-Stego achieves both high perceptual quality and visual covertness, which are critical for secure image steganography in real-world applications.

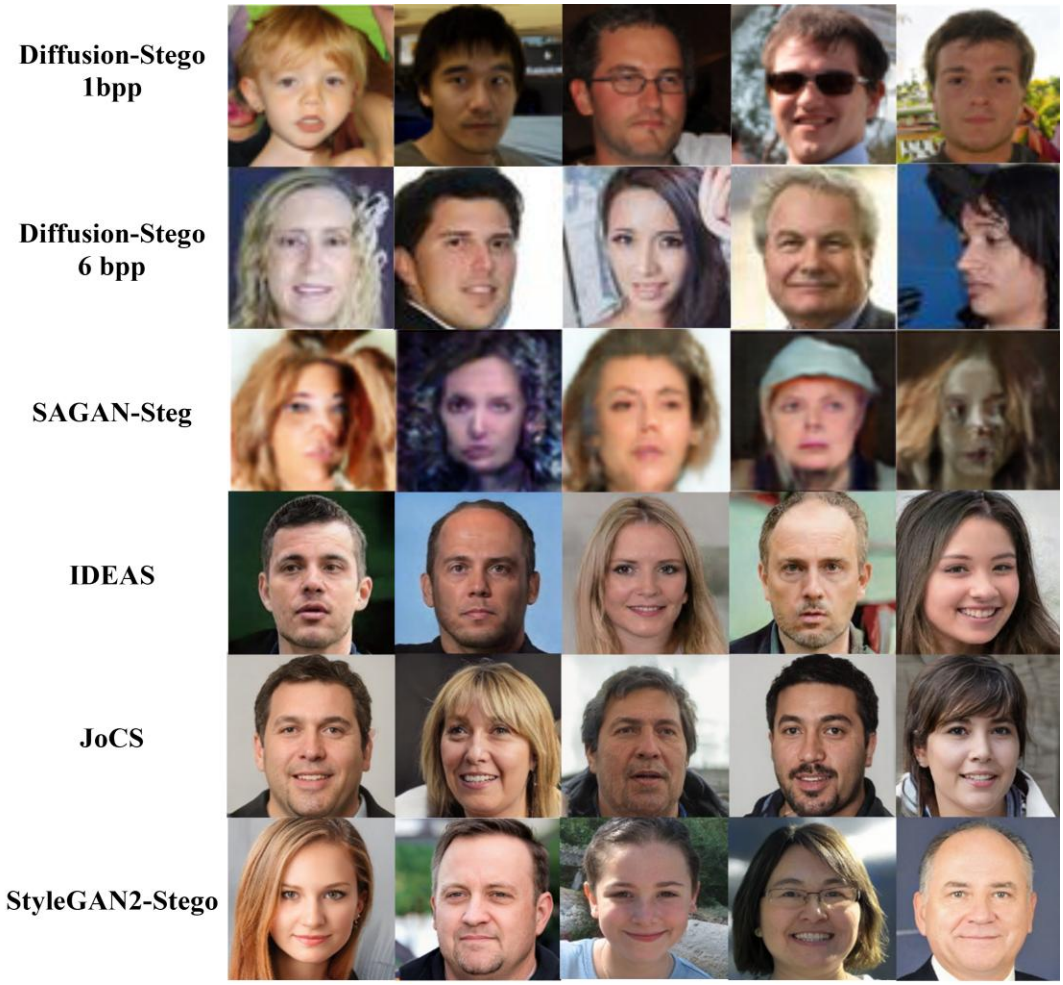

Figure 7: Stego image samples from StyleGAN2-Stego and other methods for visual quality comparison

Furthermore, to illustrate the imperceptibility of our embedding process at a payload of 0.5 bpp, we present a direct visual comparison between non-stego and stego images in Figure 8. The top row displays non-stego images generated by the pre-trained StyleGAN2-ADA, while the bottom row shows their corresponding stego versions produced by our StyleGAN2-Stego framework. The similarity between the two sets confirms that message embedding does not introduce visible distortions.

Figure 8: Visual comparison between non-stego images (top) and corresponding stego images (bottom) generated by StyleGAN2-Stego at 0.5 bpp

4.6 Overall Performance Comparison

To demonstrate the effectiveness of StyleGAN2-Stego, we compared its performance with five representative coverless image steganography methods: SAGAN-Steg [22], IDEAS [23], Zhang [28], JoCS [29], and Diffusion-Stego [30]. Table 4 summarizes the results. In this comparison, best-performing values in each column are highlighted in bold. Our proposed StyleGAN2-Stego is also bolded to emphasize its balance of high payload, strong accuracy, competitive FID, and near-optimal security.

Table 4: Payload Capacities of Compared Methods

Methods	**bpp ↑**	**Acc (%) ↑**	**FID ↓**	**Pe → 0.5**
SAGAN-Steg [22]	0.293	91.73	30.81	–
IDEAS [23]	0.0234	98.26	26.37	0.535
Zhang [28]	0.0352	**100**	–	0.497
JoCS [29]	0.00073	–	10.16	**0.5**
Diffusion-Stego [30]	1 - 6	98.12 - 91.12	**2.77 - 3.37**	0.427 - 0.385
StyleGAN2-Stego	**0.5**	**98.72**	**6.7762**	**0.494**

From Table 4, it is evident that StyleGAN2-Stego strikes an optimal balance between capacity, accuracy, and undetectability. Unlike Diffusion-Stego, which achieves slightly lower FID but suffers from reduced P_e at high payloads, StyleGAN2-Stego maintains both security and visual quality. Furthermore, the method's payload is significantly higher than SAGAN-Steg, IDEAS, Zhang and JoCS, while still delivering near-perfect extraction accuracy.

4.7 Discussion

The results show that StyleGAN2-Stego achieves a strong balance between how much data it can carry, how accurately it can be recovered, and how difficult it is to detect. At a payload of 0.5 bpp, the framework maintained a high message recovery accuracy of 98.72%. This is a notable result because many generative steganography methods lose recovery accuracy as capacity increases. In our case, the encoder–decoder design appears to learn stable and reliable latent-space adjustments, even at higher payloads.

In terms of security, the framework performed very close to the ideal case where a steganalyzer cannot distinguish between stego and non-stego images. The detection error probabilities for Xu-Net (0.4860) and Ye-Net (0.4940) are both near 0.5, meaning that these models were essentially guessing. The ROC curves back this up, showing lines close to the diagonal, which is what we expect when detection accuracy is no better than random.

For image quality, the FID score of 6.7762 confirms that the generated images remain visually similar to real FFHQ images. Although Diffusion-Stego achieves lower FID scores (2.77–3.37), its reported payload range is between 1 and 6 bpp, and results at 0.5 bpp were not provided. At its lowest reported capacity (1 bpp), Diffusion-Stego achieves strong visual fidelity but shows reduced robustness in message recovery and steganalysis resistance compared to StyleGAN2-Stego at 0.5 bpp. This highlights that prioritizing a balanced trade-off between fidelity and security, as achieved by our framework, is more practical for real-world use.

Training the encoder–decoder pair required 120 epochs on an RTX 4070 Ti GPU, which represents a notable computational cost compared to training-free approaches such as Diffusion-Stego. Nevertheless, once trained, our framework supports efficient message embedding and extraction with near real-time performance.

A likely reason for our framework's good performance is the way it works in latent space instead of directly in pixel space. By applying small, learned changes to the intermediate representation of a fixed StyleGAN2-ADA generator, the embedding process blends the hidden data into the image's high-level features. This helps maintain global image consistency and makes it harder for steganalyzers to detect any unusual patterns. That said, all results here are from clean images. In practice, images might be compressed, resized, or slightly altered during transmission, and these changes could affect recovery accuracy. Testing under these conditions is an important next step to make the method more robust for real-world use.

5. CONCLUSION

In this study, we presented StyleGAN2-Stego, a novel steganographic approach that conceals messages by applying subtle adjustments to the latent input of a pre-trained StyleGAN2-ADA generator. Unlike traditional methods that modify image contents, our framework embeds data deeper within the generative process, preserving the natural appearance of the output images.

The experimental results validate the method's effectiveness. It achieved high message extraction accuracy, strong resistance to detection by advanced steganalysis tools like Xu-Net and Ye-Net, and produced visually realistic images, as reflected by competitive FID scores. These results highlight the framework's ability to maintain a strong balance between capacity, imperceptibility, and resilience against detection.

A notable strength of StyleGAN2-Stego lies in its architecture. By keeping the generator fixed and using a streamlined encoder–decoder pair, the training process remains stable and efficient. Additionally, the system handles different message lengths with consistency, making it a practical choice for real-world use. While such frameworks can strengthen secure communication, they may also be misused for illicit purposes. This raises important ethical and security concerns, underscoring the need for counter-steganography tools, regulatory measures, and responsible use of such technologies to prevent malicious applications.

Looking ahead, future work could focus on enhancing the embedding capacity while maintaining visual fidelity and security. Other directions include improving robustness against common image transformations (compression, resizing, or scaling), and extending the framework to support additional modalities like audio or video. Potential applications include secure diplomatic and military communication, digital watermarking for intellectual property protection, embedding metadata in multimedia archives, and other domains where covert, reliable, and efficient information hiding is critical. Incorporating adversarial training may also strengthen resistance to emerging steganalysis techniques. Overall, this research underscores the promise of latent-space generative models in developing secure, high-quality tools for covert communication.

References

[1] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," Computer, vol. 31, no. 2, pp. 26–34,1998.

[2] J. Fridrich, Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge, (UK): Cambridge University Press, 2009.

[3] A. M. Qadir and N. Varol, "A Review Paper on Cryptography," in Proc. 2019 7th International Symposium on Digital Forensics and Security (ISDFS), Barcelos, (Portugal), 2019, pp. 1–6.

[4] A. Cheddad, J. Condell, K. Curran, and P. M. Kevitt, "Digital image steganography: Survey and analysis of current methods," Signal Processing, vol. 90, no. 3, pp. 727–752, 2010.

[5] J. Tao, S. Li, X. Zhang, and Z. Wang, "Towards Robust Image Steganography," IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 2, pp. 594–600, 2019.

[6] N. Subramanian, O. Elharrouss, S. Al-Maadeed, and A. Bouridane, "Image Steganography: A Review of the Recent Advances," IEEE Access, vol. 9, pp. 23409–23423, 2021.

[7] P. C. Mandal, I. Mukherjee, G. Paul, and B. N. Chatterji, "Digital image steganography: A literature survey," Information Sciences, vol. 609, pp. 1451–1488, 2022.

[8] D. R. I. M. Setiadi, S. Rustad, P. N. Andono, and G. F. Shidik, "Digital image steganography survey and investigation (goal, assessment, method, development, and dataset)," Signal Processing, vol. 206, pp. 108908, 2023.

[9] J. Kunhoth, N. Subramanian, S. Al-Maadeed, and A. Bouridane, "Video steganography: recent advances and challenges," Multimedia Tools and Applications, vol. 82, pp. 41943–41985, 2023.

[10] I. Bilal, R. Kumar, M. S. Roj, and P. K. Mishra, "Recent advancement in audio steganography," in Proc. 2014 International Conference on Parallel, Distributed and Grid Computing, Solan, (India), 2014, pp. 402–405.

[11] M. A. Majeed, R. Sulaiman, Z. Shukur, and M. K. Hasan, "A Review on Text Steganography Techniques," Mathematics, vol. 9, no. 21, pp. 2829, 2021.

[12] T. S. Reinel, R. P. Raul, and I. Gustavo, "Deep Learning Applied to Steganalysis of Digital Images: A Systematic Review," IEEE Access, vol. 7, pp. 68970–68990, 2019.

[13] N. Farooq and A. Selwal, "Image steganalysis using deep learning: a systematic review and open research challenges," Journal of Ambient Intelligence and Humanized Computing, vol. 14, pp. 7761–7793, 2023.

[14] N. J. D. L. Croix, T. Ahmad, and F. Han, "Comprehensive survey on image steganalysis using deep learning," Array, vol. 22, pp. 100353, 2024.

[15] C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," Pattern Recognition, vol. 37, no. 3, pp. 469–474, 2004.

[16] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in Proc. 2012 IEEE International Workshop on Information Forensics and Security (WIFS), Costa Adeje, (Spain), 2012, pp. 234–239.

[17] G. Xu, H. Z. Wu, and Y. Q. Shi, "Structural Design of Convolutional Neural Networks for Steganalysis," IEEE Signal Processing Letters, vol. 23, no. 5, pp. 708–712, 2016.

[18] J. Ye, J. Ni, and Y. Yi, "Deep Learning Hierarchical Representations for Image Steganalysis," IEEE Transactions on Information Forensics and Security, vol. 12, no. 11, pp. 2545–2557, 2017.

[19] V. Kumar, S. Sharma, C. Kumar, and A. K. Sahu, "Latest Trends in Deep Learning Techniques for Image Steganography," International Journal of Digital Crime and Forensics, vol. 15, no. 1, pp. 1–14, 2023.

[20] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li, "A Novel Image Steganography Method via Deep Convolutional Generative Adversarial Networks," IEEE Access, vol. 6, pp. 38303–38314, 2018.

[21] J. Li, K. Niu, L. Liao, L. Wang, J. Liu, Y. Lei, and M. Zhang, "A Generative Steganography Method Based on WGAN-GP," in Proc. 6th International Conference on Artificial Intelligence and Security (ICAIS), Hohhot, (China), 2020, pp. 386–397.

[22] C. Yu, D. Hu, S. Zheng, W. Jiang, M. Li, and Z. Q. Zhao, "An improved steganography without embedding based on attention GAN," Peer-to-Peer Networking and Applications, vol. 14, pp. 1446–1457, 2021.

[23] X. Liu, Z. Ma, J. Ma, J. Zhang, G. Schaefer, and H. Fang, "Image Disentanglement Autoencoder for Steganography without Embedding," in Proc. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, (USA), 2022, pp. 2293–2302.

[24] Z. Wang, N. Gao, X. Wang, X. Qu, and L. Li, "SSteGAN: Self-learning Steganography Based on Generative Adversarial Networks," in Proc. 25th International Conference on Neural Information Processing (ICONIP), Siem Reap, (Cambodia), 2018, pp. 253–264.

[25] Z. Zhang, J. Liu, Y. Ke, Y. Lei, J. Li, and M. Zhang, "Generative Steganography by Sampling," IEEE Access, vol. 7, pp. 118586–118597, 2019.

[26] W. Jiang, D. Hu, C. Yu, M. Li, and Z. Q. Zhao, "A New Steganography Without Embedding Based on Adversarial Training," in Proc. ACM TURC '20: Proceedings of the ACM Turing Celebration Conference - China, New York, (USA), 2020, pp. 219–223.

[27] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in Proc. NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, (USA), 2017, pp. 6629–6640.

[28] Y. Zhang, Y. Chen, H. Dou, C. Tan, Y. Luo, and H. Sang, "Image steganography without embedding by carrier secret information for secure communication in networks," PLOS One, vol. 19, no. 9, pp. e0308265, 2024.

[29] C. Ren and B. Wu, "A Robust joint coverless image steganography scheme based on two independent modules," Cybersecurity, vol. 7, no. 1, 2024.

[30] D. Kim, C. Shin, J. Choi, D. Jung, and S. Yoon, "Diffusion-Stego: Training-free Diffusion Generative Steganography via Message Projection," Information Sciences, vol. 718, pp. 122358, 2025.