Channel Attention Image Steganography With Generative Adversarial Networks

Jingxuan Tan, Xin Liao, *Senior Member, IEEE*,
Jiate Liu, Yun Cao, *Member, IEEE*, and Hongbo Jiang, *Senior Member, IEEE*

***Abstract*—Recently, extensive research has revealed the enormous potential of deep learning in the application of image steganography. However, some defects still exist in previous studies on deep learning-based steganography. In this paper, we propose a novel end-to-end network architecture for image steganography with channel attention mechanisms based on generative adversarial networks, which can yield perceptually indistinguishable stego images at various capacities. Three subnetworks constitute our model, where a generator embeds the payload into cover images, an extractor extracts it from stego images, and a powerful steganalyzer acts as a discriminator to enhance steganographic security. We design a specific channel attention module, which tunes channel-wise features in the deep representation of images dynamically by exploiting channel interdependencies. The experimental results demonstrate that the channel attention strategy is conducive to improving the quality of generated stego images and the accuracy of message extraction. To tackle the inevitable issue of extraction errors, we resort to error correction codes, with which our model achieves the maximum effective embedding rates over 4 bits per pixel. Finally, we verify that the proposed model outperforms current GAN-based steganographic schemes on two datasets and the undetectability is superior to traditional algorithms when the steganalyst cannot access model hyperparameters.**

***Index Terms*—Channel attention, end-to-end learning, generative adversarial networks (GAN), steganography.**

I. Introduction

THE boom of social networking services raises unignorable privacy protection issues [1], [2]. To prevent the disclosure of sensitive information, users can hide it in the multimedia they share, such as photos and videos. Image steganography is the art and science to conceal secret data within images [3]. Compared with cryptography, which guarantees the confidentiality of information, steganography can camouflage secret information with the content redundancy of digital media to avoid the suspicion of third parties. Thus, image steganography can be used for covert communication over public channels such as social platforms. With the image storage and computation outsourced to the cloud [4], this technology also helps achieve cloud security. As the antithesis of steganography, the core goal of steganalysis is to determine whether invisible secrets are hidden in images [5]. These two technologies have made progress in confrontation, during which many pioneering algorithms have been developed.

Early non-adaptive steganographic strategies severely damage the image statistics and often result in high detectability due to the neglect of analyzing the texture characteristics of images [6], [7]. In contrast, adaptive steganography considerably improves the statistical undetectability by making the modification to cover images concentrated in regions with edges or complex textures. Thus, adaptive steganography has become the mainstream research direction in the last decade.

At present, most successful gray-scale image adaptive steganographic algorithms are based on additive distortion functions and implemented in combination with Syndrome-Trellis Codes (STC) [8] framework that minimizes total distortion. In recent years, classical distortion functions are designed in the spatial domain [9]–[13] or in the frequency domain [11], [14], [15]. For example, HILL [12] defined the distortion function utilizing a high-pass filter and two low-pass filters. MiPOD [13] derived a closed-form expression for the detector of content-adaptive least significant bit matching based on an image model, and embedded the payload while minimizing the power of the optimal detector. Additionally, the non-additive distortion function for steganography is also developed, which harnesses the mutual embedding impacts of pixels. For instance, both CMD [16] and SMD [17] further enhance the difficulties for modern steganalysis by synchronizing the modification directions of adjacent pixels. They can be easily integrated into existing steganographic methods and utilize Gibbs [18] construction to realize the practical embedding.

The steganographic schemes above are designed for gray-scale images. When the cover image has color, it is not wise to distribute the payload evenly among the red, green, and blue channels. Following the idea in CMD [16], Tang *et al.* [19] presented an approach to clustering modification directions for

Manuscript received February 10, 2021; revised November 3, 2021; accepted December 19, 2021. Date of publication December 31, 2021; date of current version March 23, 2022. This work was supported in part by the National Natural Science Foundation of China under Grants 61972142, 61872356, U20A20181, 61732017, and 61902060, in part by National Key Technology R&D Program under Grant 2019QY(Y)0207, in part by the National Social Science Foundation of China under Grant 19ZDA103, and in part by the Hunan Provincial Natural Science Foundation of China under Grant 2020JJ4212. Recommended for acceptance by Prof. Fei Shen. *(Corresponding author: Xin Liao.)*

Jingxuan Tan, Xin Liao, Jiate Liu, and Hongbo Jiang are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: tanjx@hnu.edu.cn; xinliao@hnu.edu.cn; liujt19@mails.tsinghua.eud.cn; hongbojiang@hnu.edu.cn).

Yun Cao is with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China (e-mail: caoyun@iie.ac.cn).

Digital Object Identifier 10.1109/TNSE.2021.3139671

color images, which preserves not only the correlation within a channel but also the inter-channel interaction. Liao *et al.* [20] considered inter-channel correlations and proposed a channel-dependent payload partition strategy among RGB channels based on amplifying channel modification probabilities. Wang *et al.* discovered that G channel has a stronger correlation with R and B than the one between R and B. So they introduced a strategy that makes the modification directions of R and B channels consistent with those of G channel [21].

With the rapid development of deep learning in recent years, some steganalytic techniques exploiting convolutional neural networks have achieved satisfactory effects, such as XuNet [22], YeNet [23], Yedroudj-Net [24], Zeng-Net [25], and SRNet [26] *etc*. These deep-learning networks for steganalysis pose formidable challenges to traditional steganography. Fortunately, deep learning algorithms, especially Generative Adversarial Networks (GAN) [27], have also brought new inspiration to image steganography. GAN is generally composed of two rival subnetworks, in which the generator aims to generate data distribution approximate to the real data distribution, while the discriminator tries to distinguish real samples from the generated ones. The basic principle of GAN is to take advantage of adversarial learning to enhance the respective performance of two subnetworks. This confrontation naturally resembles the relationship between steganography and steganalysis, which implies the rationality of using GAN for developing steganographic algorithms.

According to the different functions fitted by the generator, existing popular GAN-based steganographic models can be roughly classified into four categories: coverless image steganography, generating cover images, learning distortion costs, and producing stego images. Among them, coverless image steganography via GAN learns the mapping rules of secret information to the stego image [28], [29]. Although these methods avert the irreversible disturbance caused by the modification to cover images, it is difficult to achieve high capacity and obtain high-quality images. The methods of generating cover images train the generator to produce more realistic cover images by confronting the discriminator and enhance security by learning against the steganalyzer [30]–[32]. After the training process, any traditional embedding operation can be performed on generated images. Apparently, the process of constructing images is more complex than choosing natural images as covers directly and the crackdown on fake images can also restrict its application. In distortion cost learning frameworks, the generator learns modification probability maps which can be turned into distortion costs [33]–[35]. These algorithms only substitute deep network models for traditional heuristically defined distortions, but they still rely on STC [8] to carry out the actual embedding.

The methods in [36]–[39] utilize convolutional neural networks to automatically embed and extract messages. Hayes and Danezis [36] proposed the first network architecture for image steganography through an adversarial training scheme. Zhu *et al.* [37] incorporated multiple types of optional noise layers into their model between the encoder and decoder to simulate attacks or distortions that images may experience during transmission, so their method can also be applied to digital watermarking. Zhang *et al.* [38] organized binary messages bits into three-dimensional tensors that can be concatenated with the image tensors in the encoder network, which increased the steganographic capacity significantly. Yu [39] proposed an attention based data hiding framework with GAN, in which an attention model was introduced to help the generative model to aware of inconspicuous areas of cover images. These studies show that the supervised stego image generation is a promising field of research in steganography, but they are still defective in some aspects. The stego images in [36] have severe distortion which can even be identified by human eyes. The processing operations for messages in [37] cause expensive computing complexity and storage overhead, preventing it from working normally at high capacities. The research [38] fails to resolve the contradiction between steganographic requirements and still has weakness in the anti-detection performance and decoding accuracy. The attention model in [39] generates a mask indicating the risk of causing the attention of visual detection, which can be regarded as an application of spatial attention mechanism. However, the contribution of the spatial attention mask to good imperceptibility is limited.

To address the aforementioned challenges, we propose a channel attention based end-to-end network architecture for image steganography with GAN. The model consists of three subnetworks: a generator, an extractor, and a discriminator, which are respectively responsible for generating stego images, extracting secret messages, and detecting covers and stegos. Different from the spatial attention model in [39], which tries to find suitable regions of images to hide data, we focus on the importance of each channel in the feature map. In our method, the payload is embedded in the way of fusion with the multi-channel feature maps. Since the number of meaningful features varies in different channels, each channel should be given different levels of attention. Otherwise, meaningless features may be transformed into noise that impairs the stego quality. Therefore, we introduce a channel attention unit to model channel dependencies explicitly, which enables the networks to concentrate the payload in more critical and effective channel features. Compared with the spatial attention model in [39], our channel attention module is more lightweight and effective.

It is well known that in adversarial training schemes, a well-designed discriminator can optimize itself better and give more useful feedback to the generator. In the work most relevant to ours, the discriminator is implemented by a normal convolutional neural network, which may not be beneficial enough for steganographic security. Because the discriminator plays a role similar to steganalysis, we apply a powerful gray-scale image steganalytic network, XuNet, and adjust the channel and the number of high-pass filters in the pre-processing layer as our discriminator to support color images. To overcome the problem of inevitable error rates of message extraction in the existing studies, we use error correction coding algorithms. By adding redundant parity information to the original messages we can recover them accurately from the ones disturbed by the noise. We also illustrate the effective information capacity that can be

communicated using our algorithm by virtue of a practical error-correcting scheme, Reed-Solomon codes [40].

The primary contributions of our work are as follows:

1) We propose a novel end-to-end network architecture for image steganography with channel attention mechanisms based on GAN (CHAT-GAN). The mechanism can learn channel interdependencies and adaptively adjust channel-wise features in the network activation of images, through which we can improve the quality of generated stego images and the accuracy of message extraction.
2) Our application of an advanced network for image steganalysis with the high-pass filters modified in the pre-processing layer is beneficial for enhancing security. We also use error correction codes to address the problem of inevitable extraction errors caused by the network. Combining with a ubiquitous error-correcting algorithm, we illustrate the method to estimate the effective capacity with a given payload. Under this circumstance, our model is capable of achieving effective embedding rates in excess of 4 bits per pixel while ensuring that the messages are recovered correctly.
3) Compared with the existing GAN-based methods for generating stego images, our approach brings improvement in performance measured by multiple metrics with relatively low computational complexity. The steganographic security is preferable to traditional algorithms when the model hyperparameters are unknown to the steganalyst. When generalized to another image set, our method retains its superiority.

The rest of the paper is organized as follows. In Section II, we describe the detailed network structure of the proposed CHAT-GAN, the rationale and functionality of the channel attention module, and the algorithm used for training. In Section III, we report extensive experimental results to demonstrate the performance of our model, where we also present comparisons with existing steganographic methods based on generative networks and distortion functions. Finally, we draw conclusions in Section IV.

II. Proposed Method

In this section, we present a novel end-to-end network architecture for image steganography with channel attention mechanisms based on GAN (CHAT-GAN). First, we provide an overview of the model architecture and basic ideas. Then we describe in detail the composition of each component of our model and introduce the channel attention strategy. Finally, we illustrate the loss functions and training procedures.

A. Overview of the Proposed CHAT-GAN

Our model consists of three components: (1) the generator G with parameters θ_G takes the cover image X and the message M as input and yields the stego image S; (2) the extractor E with parameters θ_E recovers the message M' from the stego image; (3) the discriminator D with parameters θ_D attempts to differentiate stegos from covers by assigning different scores to them.

The generator and the extractor both play an indispensable role in message embedding and extraction. Instead of designing encoding rules manually, generating imperceptible stego images and ensuring accurate recovery of secret information becomes the optimization objectives of the networks. In order to further improve the quality of generated images and the accuracy of message extraction, we develop a channel attention module, which tunes the deep representation of cover images dynamically by exploiting channel relationships. Since adversarial training requires the generator G and the discriminator D to be updated iteratively, the performance of the generator can be continuously improved to better fool the discriminator, achieving gradually strengthened undetectability. To this end, we adopt XuNet [22], an advanced network for steganalysis, and adjust the channel and the number of high-pass filters in the pre-processing layer as our discriminator. When training is completed, our model can be easily combined with any error correction coding algorithm to realize the actual secret communication. The upper part of Fig. 1 illustrates the overall architecture of CHAT-GAN.

B. Structure of Three Subnetworks

1) Generator: We denote a cover image with C channels and $H \times W$ size as $X \in L^{C\times H\times W}$ in which L is the range of pixel values, namely $\{0, \ldots, 255\}$. We set $C = 3$ and train our models on true-color images because they are widely used in real life. Here it is worth pointing out that our model can also be applied for gray-scale images with only a slight adjustment to the kernel configuration at the input and output layer.

For the form of the secret message, we organize it as a three-dimensional volume instead of a one-dimensional vector to achieve higher capacity. So the message $M \in \{0,1\}^{P\times H\times W}$, where P is a variable that controls the capacity. The generator G embeds the secret message into the cover image to get the stego directly, which can be represented as a function:

$$S = f_G(X, M; \theta_G). \tag{1}$$

Convolutional blocks (denoted as ConvBlocks) form the basic components of our network. Each ConvBlock includes a convolution layer (Conv), an activation function, and a Batch Normalization layer (BN). We adopt the Leaky Rectified Linear Unit (LeakyReLU) with a negative slope of 0.01 as the activation function in G. All ConvBlocks in our model have 3×3 kernels, stride 1, and padding 1.

Multiple convolution filters in the ConvBlock enlarge the number of channels of the image to learn richer features and provide more potential embedding positions. However, the features captured by the filters have only weak channel correlations. Some channels may have features that are not suitable for hiding messages. To handle this problem, we design a lightweight channel attention module (denoted as CA). It allows the network to focus on more favorable channels. We will describe it detailedly in Section II-C. Here we place it after ConvBlocks to enhance the embedding performance of the generator.

The detailed structure is shown in Table I. The form of the kernel information is "output channels $\times$(height$\times$ width$\times$input

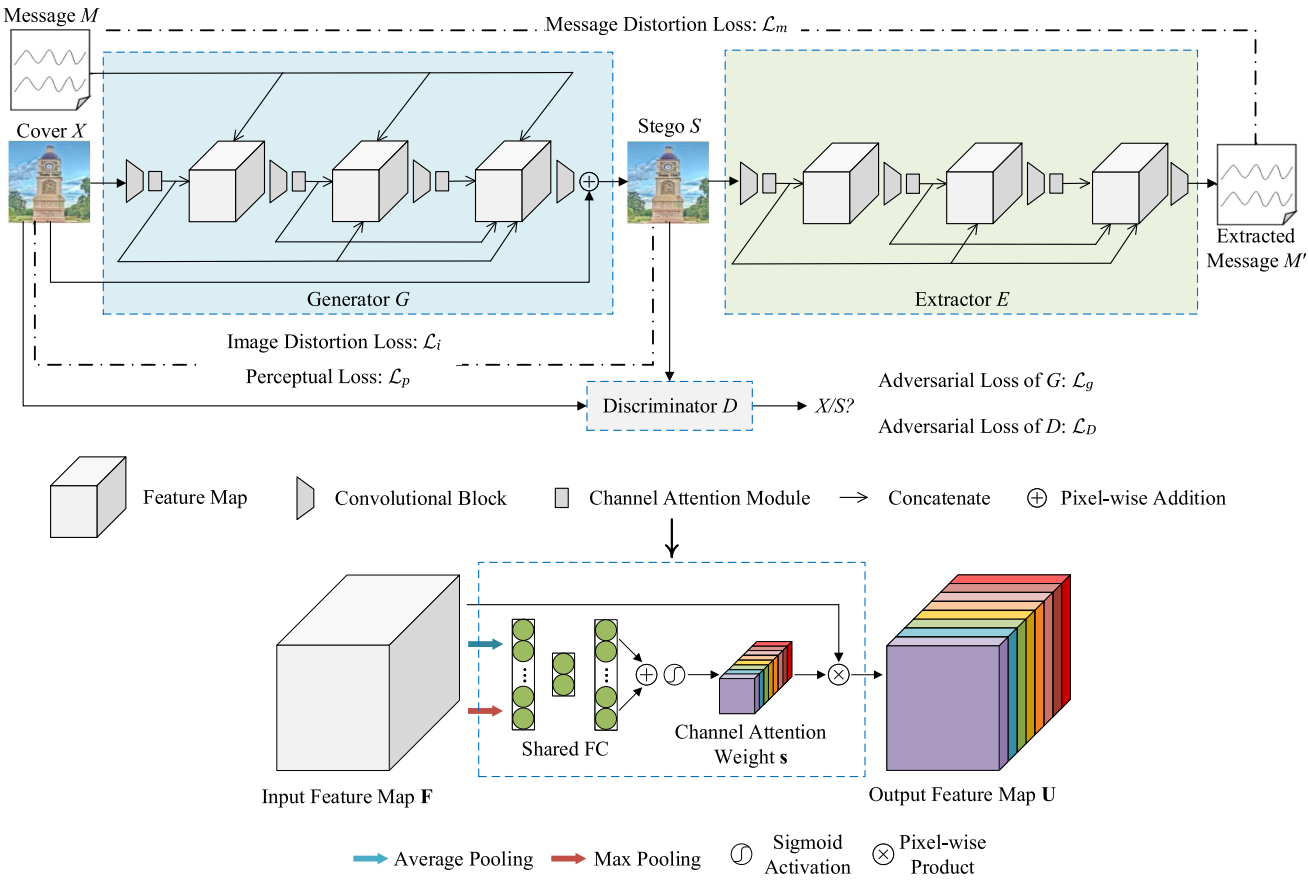


Fig. 1. An overall architecture of the proposed CHAT-GAN. The generator hides the message into the cover image and obtains the stego image. The extractor extracts the message from the stego image. The discriminator differentiates stegos from covers by assigning different scores to them. The generator and the extractor are jointly trained to minimize $\mathcal{L}_{G,E} = \lambda_i \mathcal{L}_i + \lambda_p \mathcal{L}_p + \lambda_m \mathcal{L}_m + \lambda_g \mathcal{L}_g$ while the discriminator aims to optimize $\mathcal{L}_D$. The lower part: the diagram of the channel attention module. The module calibrates the feature map **F** obtained by the convolutional block to **U** by calculating its channel weights **s**.

TABLE I
DETAILED STRUCTURE OF THE GENERATOR AND EXTRACTOR IN THE PROPOSED CHAT-GAN. THE CONVBLOCK AND CA REFER TO THE CONVOLUTIONAL BLOCK AND CHANNEL ATTENTION MODULE

	Groups	Kernel Information	Input	Output
Generator	ConvBlock-CA	$32\times(3\times3\times C)$	X	X_1
	ConvBlock-CA	$32\times(3\times3\times(32+P))$	M, X_1	X_2
	ConvBlock-CA	$32\times(3\times3\times(64+P))$	M, X_1, X_2	X_3
	Conv	$C\times(3\times3\times(96+P))$	M, X_1, X_2, X_3	X_4
	Addition	-	X, X_4	S
Extractor	ConvBlock-CA	$32\times(3\times3\times C)$	S	S_1
	ConvBlock-CA	$32\times(3\times3\times32)$	S_1	S_2
	ConvBlock-CA	$32\times(3\times3\times64)$	S_1, S_2	S_3
	ConvBlock	$32\times(3\times3\times96)$	S_1, S_2, S_3	M'

channels)". The generator consists of four convolution groups. Each convolutional layer owns 32 filters except the last, which has $C = 3$ filters, canceling the successive LeakyReLU, BN, and the channel attention unit. The cover image X passes through the first ConvBlock to get an intermediate representation, which is then explicitly tuned by the CA module with trainable parameters to get X_1. This step extracts texture features in the image preliminarily and renders more suitable embedding positions. Then the secret message M is concatenated with X_1, and processed by the second ConvBlock and CA unit to obtain X_2. Similarly, the third and fourth Conv layers take the concatenation of the message and feature maps as input, and output X_3, X_4 singly. To make full use of different levels of features, we add dense connections [41] to the third and fourth Conv layers. In other words, their inputs come from the outputs of all the preceding convolution groups. The message M is also concatenated with feature maps repeatedly to increase redundancy, which facilitates its extraction. Finally, the last convolutional layer outputs X_4. We add it to the cover X to generate the stego S. X_4 can be interpreted as the residual relative to the cover. Such residual learning makes the network training easier and helps to retain the pixel values of cover images to the maximum extent. The whole embedding process is essentially embodied as a fusion of the message volume and image features by convolutional operations.

2) Extractor: The extractor E learns to decode messages to get M' from the stego image it receives, namely:

$$M' = f_E(S; \theta_E). \tag{2}$$

The extractor has an almost identical structure to the generator. As Table I shows, ConvBlocks and CA modules are also the basic units working in the extractor. The intermediate ConvBlocks use LeakyReLU as the non-linear function, but

the last one adopts Sigmoid as activation to normalize the data to the range from 0 to 1. The dense connections are also employed in this network where the four ConvBlocks handle tensors of 3, 32, 64, 96 channels singly and finally yield the $P \times H \times W$ message tensor. The essence of message recovery is to extract information from different levels of image features.

3) Discriminator: In our model, the discriminator D scores the images. We denote the score as $f_D(I;\theta_D)$ where $I \in \{X, S\}$ represents the input image. The discriminator aims to assign higher scores to stegos and lower ones to covers.

The representation power of the discriminator will undoubtedly affect the performance of the generator because they are updated in a competitive fashion. In previous studies [37]–[39], the discriminator is simply implemented by stacking normal convolutional and fully-connected layers without particular designs. However, the discriminator faces a non-trivial problem similar to steganalysis, whose particularity lies in the necessity of extracting noise residuals. So we preliminarily deem that it may be difficult for the normal convolutional neural network to make the most of its competence as the opponent. Instead, we consider using a dedicated network for steganalysis as the discriminator.

Recent years have witnessed the successive birth of excellent steganalytic networks, among which XuNet [22], YeNet [23], and SRNet [26] are rather powerful for detecting gray-scale images in the spatial domain. They all extract image noise residuals at the beginning of the network. We use them as the discriminator in turn with slight modifications to make them capable of steganalyzing color images. We adjust the channel and the number of high-pass filters in the pre-processing layer for XuNet. We enlarge the channel of SRM [42] kernels in YeNet. The number of input channels in the first convolutional layer is altered for SRNet. The output feature numbers of the last fully-connected layer in the three networks are all modified to 1 to output the score. Preliminary experiments show that XuNet performs better so we designate it as the discriminator finally. The detailed results and analysis are provided in Section III-D2.

C. Design of the Channel Attention Module

The attention mechanism in deep learning makes the network learn to pay attention to important features and ignore irrelevant ones. In CNN-based steganography that directly generates stegos, the message is embedded in the way of fusion with the features of the cover image. The importance of these features for hiding data is different, so the attention mechanism may be useful for the performance boost. The hard attention mechanism selects a subset of the elements of the input data, discarding the rest entirely. It is typically associated with reinforcement learning because of its non-differentiability. In steganography, the information of a cover image needs to be retained as much as possible rather than discarded. In addition, the training of reinforcement learning is often inefficient. The self-attention mechanism captures internal correlations between image patches in the field of computer vision. Since the embedding operation considers the holistic image information, it has little direct contributions to steganography. We turn to the soft attention mechanism, which assigns a weight between 0 and 1 to features to indicate the level of attention that needs to be paid.

The soft attention mechanism mainly includes spatial attention and channel attention. The former allows the network to find appropriate regions of images while the latter helps to focus on favorable channels in feature maps. Ref. [39] used an attention model to generate a mask indicating the attention sensitiveness of cover images, which is a kind of spatial attention. A larger value in the mask means that the change of the corresponding pixel will lead to a higher risk of visual detection. Nevertheless, the less attention-sensitive areas shown in its paper are not complex textures or edge regions that are considered safe for adaptive steganography. Because the convolution layer itself has the effect of edge detection and texture extraction, we deem that the functionality of an additional spatial attention model is limited. This motivates us to study the effect of channel attention on image steganography.

In our approach, the convolution layer is the basic building block. It transforms the image into a multi-channel feature map, so that message bits can be hidden among these channel-wise features. When processing an input feature map, classical convolutional operations fail to capture holistic information within a channel and the dependency among channels, giving rise to some meaningless channels in the output feature map. The meaningless channels may be further turned into severe noise in output stegos, which is detrimental to stego quality and payload extraction. Thus, the important channels should be emphasized while the meaningless ones should be suppressed. To this end, we design an attention module that tunes the channels according to their importance. Taking the feature map $\mathbf{F} \in \mathbb{R}^{M \times H \times W}$ computed by the convolutional block as input, it first derives a weight vector by exploiting channel inter-dependency. Each weight reflects the importance of each channel. Then the weights are multiplied to the corresponding channels to scale the features, outputting a recalibrated feature map $\mathbf{U}$. The structure of the channel attention module is depicted in the lower part of Fig. 1.

We first aggregate spatial information in each channel of the feature map $\mathbf{F}$ using both average pooling and max pooling to obtain $\mathbf{f_{avg}} \in \mathbb{R}^{M \times 1}$ and $\mathbf{f_{max}} \in \mathbb{R}^{M \times 1}$. The m-th elements of $\mathbf{f_{avg}}$ and $\mathbf{f_{max}}$ are calculated by:

$$f_{avg}^{m} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{F}_{i,j}^{m}, \tag{3}$$

$$f_{max}^{m} = max(\mathbf{F}^{m}). \tag{4}$$

The two pooling operations compress the global information of each channel into two scalars as spatial feature statistics. To derive weights that represent the importance of each channel from these statistics, we perform linear and nonlinear operations on them. Specifically, we propagate $\mathbf{f_{avg}}$ and $\mathbf{f_{max}}$ with a shared network composed of two fully-connected layers. Next, the two output feature vectors are fused by element-wise

Algorithm 1: Training the proposed CHAT-GAN. The Generator G, the Extractor E and the Discriminator D are optimized by Adam with default hyperparameters.

Require:
The image set $\mathcal{X}$.
The initial learning rate $\alpha_1 = 0.001$ for G, E, decayed by factor 0.1 every 10 epochs. The learning rate α_2 is fixed to 0.0001 for D. The number of iterations of the discriminator per generator iteration $n_D = 5$. The batch size $b = 12$. The clipping parameter $c = 0.01$.

Ensure:
Trained networks, $\theta_G, \theta_E, \theta_D$.

while θ_G, θ_E have not converged **do**
  for $t = 1$ **to** n_D **do**
    Sample $\{X^{(i)}\}_{i=1}^{b} \sim \mathbb{P}_{\mathcal{X}}$, a batch from the image set
    Sample $\{M^{(i)}\}_{i=1}^{b} \sim$ **Bernoulli**(0.5), a batch of messages
    $g_{\theta_D} \leftarrow \nabla_{\theta_D}(\frac{1}{b}\sum_{i=1}^{b} \mathcal{L}_D(X^{(i)}, M^{(i)}))$
    $\theta_D \leftarrow \theta_D + \alpha_2 \cdot \text{Adam}(\theta_D, g_{\theta_D})$
    $\theta_D \leftarrow \text{Clip}(\theta_D, c, -c)$
  end for
  Sample $\{X^{(i)}\}_{i=1}^{b} \sim \mathbb{P}_{\mathcal{X}}$, a batch from the image set
  Sample $\{M^{(i)}\}_{i=1}^{b} \sim$ **Bernoulli**(0.5), a batch of messages
  $g_{\theta_G} \leftarrow \nabla_{\theta_G}(\frac{1}{b}\sum_{i=1}^{b} \mathcal{L}_{G,E}(X^{(i)}, M^{(i)}))$
  $g_{\theta_E} \leftarrow \nabla_{\theta_E}(\frac{1}{b}\sum_{i=1}^{b} \mathcal{L}_{G,E}(X^{(i)}, M^{(i)}))$
  $\theta_G \leftarrow \theta_G + \alpha_1 \cdot \text{Adam}(\theta_G, g_{\theta_G})$ $\theta_E \leftarrow \theta_E + \alpha_1 \cdot \text{Adam}(\theta_E, g_{\theta_E})$
end while

addition, after which the fused feature vector is transformed to a channel weight vector via the Sigmoid function. Namely, the weight vector $\mathbf{s} \in \mathbb{R}^{M\times 1}$ is obtained by:

$$\mathbf{s} = \sigma(\mathbf{W_2}(\delta(\mathbf{W_1}\mathbf{f_{avg}})) + \mathbf{W_2}(\delta(\mathbf{W_1}\mathbf{f_{max}}))), \tag{5}$$

where δ, σ denote the Rectified Linear Unit and Sigmoid function respectively, $\mathbf{W_1} \in \mathbb{R}^{M/r\times M}, \mathbf{W_2} \in \mathbb{R}^{M\times M/r}$ refer to the weights in each layer. It is noted that the hidden layer performs dimensionality reduction on the inputs to balance the performance and computational complexity of the model. We set the reduction factor $r = 16$, referring to the optimal configuration in a related attention mechanism study [43]. Finally, each element of $\mathbf{s}$, as a scalar, is multiplied by each channel of $\mathbf{F}$ to calculate $\mathbf{U} = [\mathbf{U}^1, \mathbf{U}^2, \ldots, \mathbf{U}^M]$. The computation of the m-th channel of $\mathbf{U}$ can be expressed as:

$$\mathbf{U}^m = s^m \mathbf{F}^m. \tag{6}$$

In this way, useless channels are suppressed by multiplying with lower weights. And vice versa. After modifying channels, $\mathbf{U}$ has a stronger representation ability for stego generation or message recovery.

D. Loss Functions and Algorithm

Since the generator and the extractor work end-to-end in a pipeline, we update them synchronously. Based on the idea of mutual confrontation, the discriminator is optimized alternately with them. The fundamental requirement of steganography is to guarantee the visual indistinguishability between the cover X and the stego S. To achieve this goal, we take the mean square error to indicate the image distortion loss:

$$\mathcal{L}_i = MSE(X, S) = \frac{1}{C \times H \times W}\|X - S\|_2^2. \tag{7}$$

Furthermore, deep network activation has been shown to work well as an indicator of perceptual loss when training neural networks that produce images as output [44], [45]. We feed the cover X and the stego S to a pre-trained VGG [46] and minimize the difference between their feature maps at several depths. This perceptual loss is calculated by:

$$\mathcal{L}_p = MSE(VGG(X), VGG(S)). \tag{8}$$

Steganography also requires accurate recovery of secret messages from the stego. Each value in the original message M is either 0 or 1, while each element of the extracted message M' is a floating point number in the range of 0 to 1. In training, we employ the binary cross entropy loss to minimize the distinction between M' and M:

$$\mathcal{L}_m = \frac{1}{PHW}\sum_{i=1}^{PHW} -M_i \log_2 M_i' - (1 - M_i)\log_2(1 - M_i'). \tag{9}$$

When training is completed and the model is practically applied, M' needs to be rounded to 0 or 1 to construct the real bit sequence.

A perfectly secure steganographic system requires that stegos and covers follow the same distribution. If we measure the distance between their distributions using the KL divergence, the divergence value D_{KL} is equal to 0. However, it is difficult to achieve perfect security when implementing actual stegosystems, but we can optimize them from this perspective. The original GAN proposed by Goodfellow *et al.* can optimize the generated distribution on the basis of KL divergence [27]. Subsequent researchers are committed to improving GAN by constructing suitable network structures or proposing new loss functions to eliminate its various defects. Among them, Arjovsky *et al.* found that Wasserstein distance has advantages cover KL or JS distance and proposed Wasserstein-GAN (WGAN) to stabilize training [47]. Therefore, we employ WGAN to better fit the generator output distribution, which can also achieve the effect of stable training. In steganography, the Wasserstein-1 distance of cover distribution $\mathbb{P}_{\mathcal{X}}$ and stego distribution $\mathbb{P}_{\mathcal{S}}$ is expressed as:

$$W(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_{\mathcal{S}}) = \inf_{\gamma\in\Pi(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_{\mathcal{S}})} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|], \tag{10}$$

where $\Pi(\mathbb{P}_{\mathcal{X}}, \mathbb{P}_{\mathcal{S}})$ denotes the set of all possible joint distributions $\gamma(x, y)$ whose marginals are respectively $\mathbb{P}_{\mathcal{X}}, \mathbb{P}_{\mathcal{S}}$. Given a joint distribution, $\mathbb{E}$ refers to the expected value of the l_1 distance between cover and stego pairs sampled from it. In all possible joint distributions, the infimum on the expected values is defined as the Wasserstein-1 distance. However, the infimum is

difficult to solve directly. Ref. [47] thus transforms it appropriately to an optimizable loss. Under the guidance of its method, the discriminator assigns different scores to real and generated samples. We use the output result of the discriminator on the stego image as the adversarial loss of generator. It can be expressed as:

$$\mathcal{L}_g = f_D(S;\theta_D). \tag{11}$$

In summary, the objective of generator/extractor network is to minimize:

$$\mathcal{L}_{G,E} = \lambda_i\mathcal{L}_i + \lambda_p\mathcal{L}_p + \lambda_m\mathcal{L}_m + \lambda_g\mathcal{L}_g, \tag{12}$$

where $\lambda_i, \lambda_p, \lambda_m, \lambda_g$ control the relative weight of each item. We will discuss how to set them in Section III-B.

The discriminator endeavors to diminish the prediction score on covers and magnify it on stegos. We update it through the following loss:

$$\mathcal{L}_D = f_D(X;\theta_D) - f_D(S;\theta_D). \tag{13}$$

The training procedure is described in Algorithm 1. The generator/extractor and discriminator are trained alternately until the loss can converge, where G and E jointly learn to minimize $\mathcal{L}_{G,E}$ while D aims to minimize $\mathcal{L}_D$. Note that D is iterated 5 times once G is iterated. θ_D should be truncated to $[-0.01, 0.01]$ after each update to satisfy the Lipschitz continuity [47]. In this mode, the generator can attain reliable gradients to boost the performance of steganography continuously.

III. Experimental Results and Analysis

In this section, we first introduce several criteria among different aspects to assess our model. Then the experimental setup and hyperparameter selection are illustrated. To verify the effectiveness of the channel attention modules and the discriminator, we conduct the ablation study. Then the impact of different structures of the channel attention unit and discriminator is further studied. We also explore the feasibility of embedding with various payloads and the effective capacity with error correction codes. Next, we show our comparisons with existing stego generation frameworks and distortion minimization frameworks for image steganography. We additionally consider the generalization of our model and test it on a different image set. We end this section by validating the performance of our method on other distributions of payloads and discussing the computational efficiency.

A. Criteria to Evaluate the Performance

We evaluated our steganographic systems from the following four aspects: *image quality*, the similarity between covers and stegos; *security*, the ability to resist detection by steganalysis; *extracting accuracy*, the resemblance between original messages and the extracted ones; *capacity*, the amount of data that can be hidden in an image.

1) Image Quality:

- Peak Signal to Noise Ratio (PSNR) [48]: PSNR is one of the most commonly used objective criteria for evaluating images, which is defined via mean square error (MSE). Given a distortion-free color image X of size $C \times H \times W$ and its noisy approximation Y, MSE is calculated by:

$$\mathrm{MSE} = \frac{1}{CHW}\sum_{i=1}^{C}\sum_{j=1}^{H}\sum_{k=1}^{W}(X_{i,j,k} - Y_{i,j,k})^2. \tag{14}$$

  Then PSNR is defined as:

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{MAX_I^2}{\mathrm{MSE}}\right), \tag{15}$$

  where MAX_I is the maximum possible pixel value of the image. A larger value of PSNR usually means less distortion.
- Structural Similarity (SSIM) [49]: SSIM is based on three comparative measures between two samples: luminance, contrast, and structure. SSIM of two images X and Y can be obtained as follows:

$$\mathrm{S}SIM = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_{XY} + c_2)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)}, \tag{16}$$

  where $\mu_X, \mu_Y, \sigma_X^2, \sigma_Y^2$ are the means and the variances of X and Y respectively, and σ_{XY} is the covariance. $c_1 = (k_1\ L)^2$ and $c_2 = (k_2\ L)^2$ are constants used to maintain numerical stability to avoid zero in the denominator, where L is the dynamic range of pixel values and $k_1 = 0.01, k_2 = 0.03$. Following the rules in [49], we used a sliding window with stride 1 and size 11×11 to run across the whole image and calculated the values of SSIM between two image patches each time. The mean value of each image patch is a weighted average of the pixel value and its corresponding weight. The weights applied to an N-pixel image patch $\mathbf{w} = \{w_i | i = 1, 2, \ldots, N\}$ are generated by a circular-symmetric Gaussian function with standard deviation of 1.5 and normalized to unit sum such that $\sum_{i=1}^{N} w_i = 1$. If the weight corresponding to pixels x_i and y_i is expressed as w_i, then the mean $\mu_X = \sum_{i=1}^{N} w_i x_i$, variance $\sigma_X^2 = \sum_{i=1}^{N} w_i(x_i - \mu_X)^2$, and covariance $\sigma_{XY} = \sum_{i=1}^{N} w_i(x_i - \mu_X)(y_i - \mu_Y)$. This calculation is performed on gray-scale image patches. Finally, we averaged all results in three channels to get the mean SSIM. SSIM ranges from 0 to 1. A higher value indicates a stronger perceived similarity between the two images.
- Perceptual Loss ($\mathcal{L}_p$): Some traditional image quality evaluation criteria, such as PSNR and SSIM, are shallow functions that may fail to account for many nuances of human perception in some cases [44]. We also took the perceptual loss gained via deep network activation in a

pre-trained VGG (in (8)) as a metric to evaluate image qualities. Lower values imply that the stego images are more perceptually consistent with the cover images.

2) Security: The security of stenography is usually empirically assessed by statistical undetectability against steganalysis. When using a certain steganalytic algorithm to detect images generated by a steganographic scheme, the detection error can reflect security. It is the average of the false-alarm rate (recognizing covers as stegos) and the missed-detection rate (recognizing stegos as covers). A larger detection error means stronger security. We evaluated the security of our model through two steganalytic techniques. The first detector we used is a classic rich model, SCRMQ1 [50], which is a high-dimensional feature extractor for color images in the spatial domain. It works well with an ensemble classifier [51] to detect images. Another steganalyzer we adopted was WISERNet [52], a state-of-the-art convolutional neural network dedicated to steganalysis of color images. It is distinct from the discriminator in our network, which can show the anti-detection ability against the steganalyzer not seen during training.

3) Extracting Accuracy: The extracting accuracy of messages is also one of the essential requirements in steganography. Since the outputs of the extractor are floating point values in the range [0,1], we rounded them to $\{0, 1\}$ for computing the accuracy. It is calculated as the same number of bits between M and M' divided by the total number of bits. Higher accuracy means that the predicted M' is closer to M. In tables of this paper, we refer to the message extraction accuracy as accuracy for brevity.

4) Capacity: Bits Per Pixel (bpp) is a general measure of embedding capacity, which indicates how many bits of information can be held in each pixel of an image. The embedding capacity usually contradicts security. In general, a payload with lower capacity will cause less distortion to the image, leading to higher security. In our method, P reflects the embedding capacity. We add error-correcting characters after messages for practical use. In this case, we should know the number of *effective* bits per pixel. A method to derive the effective capacity will be analyzed in Section III-E.

B. Experimental Setting

We used the platform of Pytorch to implement our network. Models were trained on 20,000 images from the COCO [53] dataset and 1,000 images were used for validating. The one with the best validation performance was evaluated on the test set of 1,000 images. The images in the training set were randomly cropped and then scaled to 256×256 by bilinear interpolation, after which random flips were conducted before they were passed to the generator. These transforms were devised to enhance the robustness of our model. The tests were performed on 256×256 cropped images without transforms. Each bit of a message was sampled from the Bernoulli distribution with a probability of 0.5. The Adam [54] optimizer was utilized with default hyperparameters except for the learning rate settings. As is revealed in Algorithm 1, the generator and the extractor took the learning rate decay policy while the discriminator fixed it during the training process. All deep-learning based programs, including the compared algorithms and the CNN for steganalysis, run on Geforce RTX 2080 Ti GPU.

TABLE II
PERFORMANCE WITH THE DIFFERENT SETTINGS OF RELATIVE WEIGHT OF EACH ITEM IN THE LOSS FUNCTION $\mathcal{L}_{G,E}$

Criteria	$\lambda_i, \lambda_p, \lambda_m, \lambda_g$				
	1, 1, 100, 1	**1, 1, 1000, 1**	**1, 0.1, 100, 1**	**1, 1, 100, 100**	**1, 0.1, 100, 100**
PSNR	46.19	43.71	45.97	46.48	45.09
SSIM	0.9942	0.9910	0.9940	0.9944	0.9933
$\mathcal{L}_p$	15.20	24.34	17.92	14.54	24.80
Accuracy	99.10%	99.71%	99.76%	99.03%	99.43%

In (12), λ_i, λ_p strengthen the stego image quality, λ_g emphasizes the undetectability, while the message recovery accuracy is underlined by λ_m. We first set them all to 1 to train the network to embed 1 bpp payload. It was observed that the losses could not converge. PSNR and SSIM were kept at 100 and 1 respectively, but the message extraction accuracy was maintained at around 50%. It indicated that the network completely failed to extract the message, so we increased λ_m to 10. However, the problem of loss convergence failure still arose. When λ_m was set to 100, the model converged successfully. We further tried to raise λ_m to 1000 but found a marked reduction in stego image quality. Hence we fixed $\lambda_i = 1, \lambda_m = 100$ and proceeded to adjust λ_p, λ_g. At that point, the orders of magnitude of $\lambda_i\mathcal{L}_i$, $\lambda_p\mathcal{L}_p$, $\lambda_m\mathcal{L}_m$, $\lambda_g\mathcal{L}_g$ were 10^0, 10^1, 10^0, 10^{-2} respectively. We altered λ_p to 0.1 and λ_g to 100 in turn, making the weighted item reach the order of 10^0. Table II gives the performance under five combinations of the relative weights. The results have shown that there exists a trade-off between stego image quality and message recovery accuracy. Since $\mathcal{L}_p$ measures image quality, the reduction of λ_p weakens the concealment, *i.e.* lower PSNR, but leaves more traces that help to recover messages, *i.e.* higher extracting accuracy. Similarly, the increase of λ_g brings about higher image quality but lower extraction accuracy. To achieve a good trade-off, we set $\lambda_i = 1, \lambda_p = 1, \lambda_m = 100, \lambda_g = 1$ in subsequent experiments.

In traditional spatial image steganography, the change of pixel value can only be $+1$, -1 or 0 when embedding a payload. Our method is different from this kind of ± 1 embedding because the pixel value change is equal to the residual outputted by the last convolution layer in the generator. The residual has positive and negative values representing different modification directions, but their absolute values are all greater than 1. To see the effect of limiting the residual to a certain range, we truncated the residual. As shown in Table III, when the truncation factor is 1, the generator conducts ± 1 embedding. Similarly, we set it to 5, 10, and the ∞ means no truncation. It can be found that with the truncation factor descending, the stego image quality is markedly boosted but the extracting accuracy is compromised. Again, we intended to achieve a proper trade-off and took 10 as the factor. Using this configuration, the convergence plots of PSNR and extraction accuracy of our model when embedding 1 bpp payload are presented in Fig. 2. It

TABLE III
PERFORMANCE WITH THE DIFFERENT TRUNCATION FACTORS IN THE GENERATOR

Criteria	Truncation Factor			
	1	**5**	**10**	∞
PSNR	50.83	46.69	46.42	46.19
SSIM	0.9972	0.9945	0.9943	0.9942
$\mathcal{L}_p$	10.10	14.22	14.82	15.20
Accuracy	91.39%	98.75%	99.07%	99.10%

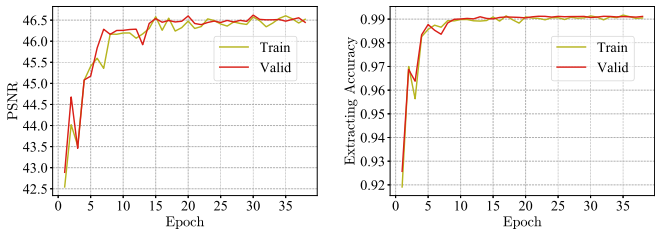


Fig. 2. The PSNR (left) and extraction accuracy (right) convergence plots of our proposed network when embedding 1 bpp payload.

shows that the network performance improves rapidly in earlier epochs and gradually converges in later ones.

C. Ablation Study of the Channel Attention Unit and Discriminator

We introduce channel-wise attention in our networks to refine the payload distribution dynamically. The discriminator in our model is used for boosting the training of the generator to keep its output gradually approximates the distribution of covers. To investigate the effectiveness of channel attention modules and the discriminator, we performed an ablation study by varying the configuration in the proposed CHAT-GAN. Then we got two variants as follows: (1) Variant #1: removing channel attention modules but retaining the discriminator; (2) Variant #2: removing the discriminator but retaining channel attention modules. We trained CHAT-GAN and its variants with 1 bpp and tested their performance on image quality and extracting accuracy. When Variant #2 was trained, the adversarial loss in (11) was not included in the total loss in (12) due to the disuse of the discriminator.

The results are shown in Table IV. The comparison between CHAT-GAN and Variant #1 illustrates that channel attention modules bring considerable improvement on all evaluation criteria. These modules ensure more useful features are emphasized and the meaningless ones are suppressed, producing stegos with less noise. Due to the symmetric application of attention blocks in the extractor, it is also beneficial for recovering messages more precisely. After canceling the discriminator, slight performance degradation occurs in Variant #2. This proves that the discriminator can provide reliable feedback for the generator, which forces it to enhance the imperceptibility of stego images. The message extraction accuracy is also influenced by the discriminator because the generator and the extractor are trained synchronously.

TABLE IV
ABLATION STUDY TO COMPARE OUR PROPOSED CHAT-GAN AND THE VARIANTS

Criteria	CHAT-GAN	Variant #1	Variant #2
PSNR	46.42	45.37	46.08
SSIM	0.9943	0.9924	0.9937
$\mathcal{L}_p$	14.82	16.29	14.91
Accuracy	99.07%	98.68%	98.79%

TABLE V
IMPACT OF DIFFERENT CHANNEL ATTENTION MODULES

Criteria	SENet [43]	GSoP [55]	ECA [56]	CHAT-GAN
PSNR	46.13	46.27	46.22	46.42
SSIM	0.9939	0.9936	0.9942	0.9943
$\mathcal{L}_p$	15.98	16.02	15.74	14.82
Accuracy	99.03%	99.00%	99.01%	99.07%

D. Impact of Different Designs of the Channel Attention Unit and Discriminator

The channel attention unit and discriminator can be constructed using other architectures. We studied the design of the structure of attention units, replaced ours with similar work, and gave comparative results. We also provided the best choice of discriminator by considering model performance and complexity comprehensively.

1) Impact of Different Channel Attention Units: The results of the ablation study manifest the advantage of the attention mechanism. We further explored the state-of-the-art attention modules available till now. The Squeeze-and-Excitation (SE) block [43] simply applies average pooling in the squeeze phase and uses fully-connected layers to excite attention. Gao *et al.* [55] introduced a global second-order pooling strategy (GSoP) for more effective feature aggregation, which is used for tensor scaling along channel dimension after nonlinear transformation. The efficient channel attention (ECA) module proposed by Wang *et al.* [56] avoids dimensionality reduction and captures cross-channel interaction in an efficient way. It uses 1D convolution to convert the average-pooled features into attention weights.

We replaced our attention module with SE, GSoP, and ECA in turn. It was found that GSoP required the highest GPU memory and took the longest training time. The results are reported in Table V. It can be observed that the proposed attention module performs better than others. This may be because we use both average and max pooling to aggregate global information compared with SE and ECA. In this way, more activation state information of the cover image is preserved, so more refined weights are derived. The complexity of GSoP exceeds even that of the generator and extractor due to the calculation of higher-order statistics, which is the worst option.

2) Impact of Different Discriminators: Since the discriminator needs to distinguish between covers and stegos, we naturally thought of making a network for steganalysis play this role. However, existing stego generating frameworks

TABLE VI
IMPACT OF DIFFERENT ARCHITECTURES OF THE DISCRIMINATOR

Criteria	YeNet [23]	SRNet [26]	Critic [38]	CHAT-GAN
#Param	109 K	4778 K	20 K	15 K
GFLOPs	1.932	5.340	1.259	0.100
PSNR	46.13	46.32	46.25	46.42
SSIM	0.9942	0.9942	0.9943	0.9943
$\mathcal{L}_p$	14.81	14.56	14.50	14.82
Accuracy	99.07%	99.08%	99.03%	99.07%

TABLE VII
PERFORMANCE OF THE PROPOSED CHAT-GAN WITH DIFFERENT PAYLOADS

Criteria	Capacity				
	1 bpp	2 bpp	3 bpp	4 bpp	5 bpp
PSNR	46.42	43.17	41.84	38.92	36.93
SSIM	0.9943	0.9880	0.9832	0.9668	0.9458
$\mathcal{L}_p$	14.82	27.16	47.72	232.74	441.66
Accuracy	99.07%	97.46%	96.18%	94.56%	92.36%
Effective Capacity	0.9814	1.8984	2.7708	3.5648	4.2360
Code Rate	98.14%	94.92%	92.36%	89.12%	84.72%

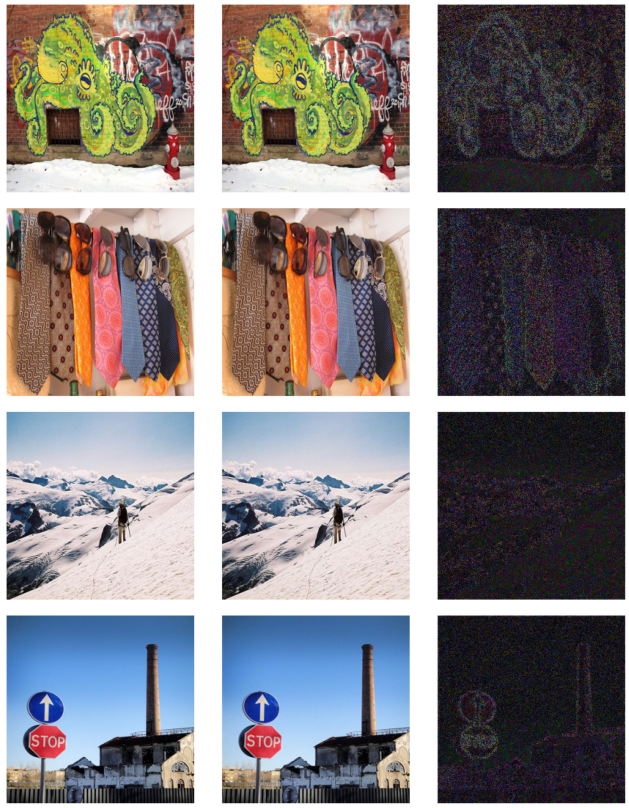
in [37]–[39] use normal convolutional neural networks to confront the generator. To investigate the efficacy of different discriminators, we also substituted a modified YeNet [23], SRNet [26] and Critic [38] for our discriminator. The comparative results with different discriminators are given in Table VI. As can be seen, they differ little in performance but XuNet has the fewest parameters and floating point operations. YeNet may have encountered the vanishing gradient problem due to its great depth. SRNet addresses it through residual learning. Unexpectedly, the Critic using a regular CNN is not inferior. To sum up, XuNet matches the generator and the extractor best in terms of both performance and computational cost, that is why we designate it as the discriminator finally.

E. Practical Application

Although embedding capacity conflicts with other metrics, a good steganographic system demands flexibility and availability at various payloads. We trained our models with higher payloads by varying the data depth P from 1 to 5. Table VII reveals that with the increase of the payload, all the indicators get worse. This is because the growth in the amount of embedded information will inevitably cast more noise on images, thus decreasing their quality.

Fig. 3 shows some cover images in the test set and the corresponding stego images with 5 bpp. It can be seen that even when the payload grows to 5, our model can still yield visually indistinguishable images. It leaves no artifacts even though the images have monotonous colors or relatively few textures. The difference between cover and stego images also demonstrates that our network embeds information at complex textures or edge regions.

In previous experiments, each bit of the message was randomly generated during training and testing. The models



(a) cover (b) stego (5 bpp) (c) difference

Fig. 3. Covers (a) and stegos with 5 bpp payload (b) generated by the proposed method. The per-pixel difference (c) is magnified 5 times for clearer visualization. Brighter parts correspond to areas with complex textures or edges in covers.

trained with less than 4 bpp payload have an extracting accuracy of more than 96%. However, in actual scenarios, secret messages are usually encrypted before embedding into the images, so it only makes sense to recover the message with 100% accuracy before decryption. The same challenge exists in other GAN-based methods in which the stegos are generated directly by the network [37]–[39]. A primary reason is that neural networks have always struggled to achieve 100% accuracy. To address this issue, we can employ error correction codes.

By adding redundant information to the original message before embedding, the coding scheme ensures it can be precisely recovered from the extracted one by the network. But the additionally added redundancy will reduce the effective capacity. We take Reed–Solomon codes [40] (referred to as RS codes) as an example to illustrate its utility and effective capacity. We denote a Reed-Soloman codec working in the Galois Field of 2^s as $RS(n,k)$ with s-bit characters. This means the encoder treats s bits as a character, initializes original messages to a k-character block, and expands the block to n characters. So there are $n-k$ error-correcting characters in the final block. RS can correct errors and erasures at the same time, up to a limit of $2\times e+v\leq(n-k)$, where e is the number of errors and v is the number of erasures. Steganography usually assumes that the images will not suffer distortion such as compression, noise, blur, and geometric attacks in the

public channel. Hence the only source of distortion is the extracting error of the neural network, whose error probability is $\epsilon = 1 - ext\_acc$ (ext_acc refers to the extracting accuracy of messages). We simulated our model as a binary symmetric channel with no erasures, so the maximum number of allowable errors is $\frac{n-k}{2}$. Given the error probability ϵ and the total capacity $n = P$, we let $n \times \epsilon \leq \frac{n-k}{2}$ to meet the RS decoding condition. Therefore, the maximum effective capacity is $k = (2 \times ext\_acc - 1)P$ bpp. Table VII also shows the maximum effective capacity and code rate (k/n) if RS codes are used. As can be seen from the results, these two criteria decrease as the capacity grows, but the maximum effective embedding capacity still exceeds 4 bpp.

We implemented a practical steganographic system by combining our trained model and Reed-Soloman codes, which can embed text information into images and extract it perfectly correctly. The Reed-Soloman codec we used worked in GF(2^8) with 4 error-correcting characters. Suppose the secret message is "Hello, world!". At the sender's end, the RS encoder was placed in front of the trained generator model to append error-correcting characters to the original message. Here the 4 error-correcting characters are "\xc4f \ x18\xb7". The entire block containing original characters and error-correcting ones was converted to a binary sequence, which was then replicated to fill a map of the same size as the cover as the input of the generator. At the receiver's end, we rounded the output of the extractor to get an extracted message map and converted the duplicate binary fragments back into multiple byte arrays. The RS decoder processed each byte array separately so that numerous correct messages can be produced ultimately.

We provided an example to explain how the RS decoder repairs a damaged message. Suppose one of the extracted byte arrays is "Lello, world)\xc4f \ x18\xb7", where there are two error characters that need to be repaired. Since the RS codec treats a message as a polynomial in the Galois Field, the error correcting procedures involve polynomial computations that follow rules of finite field arithmetic. The key step to locate error characters is the calculation of the error locator polynomial using the Berlekamp-Massey algorithm. The positions of error characters are derived from the roots of this polynomial. The critical step to correct the error is the computation of error magnitude using the Forney algorithm. In this example, the error magnitude is a 17-character byte array, where the 1st and 13th positions are "\x04" and "\x08" and other values are 0. By subtracting the error magnitude from the input damaged message, we can get the correct one.

F. Comparative Experiments

1) Comparisons With Stego Generating Frameworks: The GAN-based methods most relevant to ours include HiDDeN [37], SteganoGAN [38], and ABDH [39]. HiDDeN jointly trains encoder and decoder networks to encode information into images and recover information from images. Their model can be made robust by training against multiple types of noises. SteganoGAN improves the structure of encoder and decoder, enabling higher payloads to be hidden in an image. ABDH consists of two discriminator networks, one to distinguish between cover and stego images, and the other to distinguish between original secret payloads and the extracted ones. An attention model is introduced to facilitate the generator to produce better stego images without perturbations of the spotlights. Furthermore, it devises the inconsistency loss to ensure that the payloads can not be extracted from the cover images. We compared CHAT-GAN with these approaches in this section. The models were all trained and tested on the same image set like ours.

It is noticed that the HiDDeN model can not be directly trained on 256 $\times$ 256 images to match our embedding capacity. For example, to train a model with 1 bpp, we need to sample a message vector of 65,536 bits and replicate it spatially 65,536 times to concatenate with the image activation volume. To reduce the memory overhead and computational cost, we trained HiDDeN on 16 $\times$ 16 cropped images to encode L-bit chunk of the message, which followed the setup in its paper. We varied $L = 256$, 512, 768 to match the capacity of 1 bpp, 2 bpp, 3 bpp. The tests were still conducted on 256 $\times$ 256 images, where encoding and decoding were performed on each 16 $\times$ 16 patch. The framework of ABDH is mainly designed for digital watermarking to hide another image in an image. To make it available for steganography, we replaced the secret image with random bits. Although the authors used the training data mixed with noisy samples to enhance the robustness of the watermark, we used noise-free images to train their network since steganography assumes that images are protected from attack.

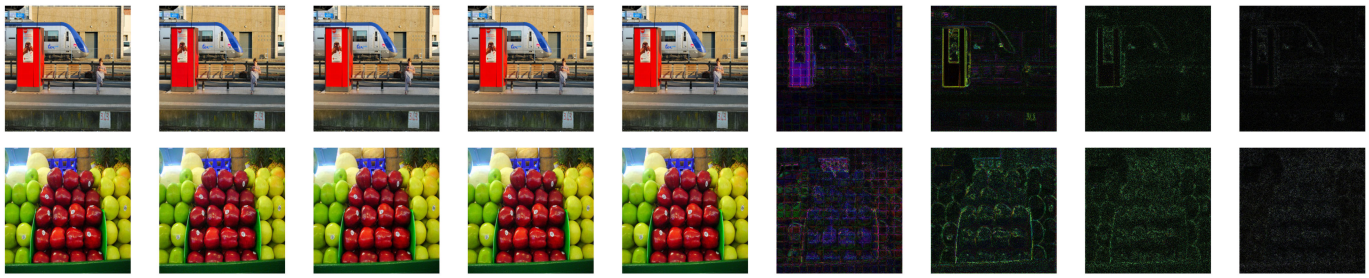
To evaluate steganographic security, traditional and deep learning-based detectors were both taken into consideration. For traditional detection schemes, we created 2,000 pairs of covers and stegos from COCO and extracted features through SCRMQ1 [50]. Then half of the pairs were used for training an ensemble classifier [51] and the other half formed the test set. The tests were run ten times. In terms of CNN-based steganalysis, we trained WISERNet [52] with 5,000 pairs of covers and stegos. Ten trained models were tested on 1,000 pairs in each experiment. The average detection error for ten test runs by the ensemble classifier (denoted as Detection Error #1) and WISERNet (denoted as Detection Error #2) was reported to assess security performance. From a practical point of view, the maximum effective capacity when using Reed-Soloman codes was also compared. Traditional metrics to evaluate the quality of synthesized images, such as PSNR which is based on per-pixel mean square error, do not assess joint statistics of the results [57], [58]. Thus, we selected the Inception Score (abbreviated as IS) [59] as an evaluation metric, which uses an Inception v3 Network pre-trained on ImageNet to calculate a statistic of the generated images concerning quality and diversity.

The performance comparisons between CHAT-GAN and other GAN-based steganographic algorithms [37]-[39] with various payloads on COCO dataset are shown in Table VIII. Compared with HiDDeN, at 1 bpp, the performance enhancement of CHAT-GAN mainly focuses on message extraction accuracy and effective capacity. In particular, the effective capacity of our model is almost 6 times that of HiDDeN. This is mainly because its decoder ends up using a fully-connected

TABLE VIII
COMPARISONS BETWEEN CHAT-GAN AND OTHER GAN-BASED STEGANOGRAPHIC ALGORITHMS WITH VARIOUS PAYLOADS ON COCO DATASET

Capacity	Method	PSNR	SSIM	$\mathcal{L}_p$	Accuracy	Effective Capacity	Detection Error #1	Detection Error #2	IS
1 bpp	HiDDeN [37]	42.07	0.9876	585.91	58.28%	0.1656	0.0003±0.0005	0.0002±0.0003	21.68
	SteganoGAN [38]	42.52	0.9894	268.50	97.75%	0.9550	0.0005±0.0003	0.0008±0.0006	21.91
	ABDH [39]	43.21	0.9872	162.76	97.50%	0.9500	0.0006±0.0005	0.0005±0.0010	21.78
	CHAT-GAN	46.42	0.9943	14.82	99.07%	0.9814	0.0027±0.0008	0.0024±0.0039	21.82
2 bpp	HiDDeN† [37]	48.36	0.9972	170.92	50.00%	0.0000	0.0108±0.0015	0.0205±0.0118	21.85
	SteganoGAN [38]	39.68	0.9801	538.34	96.46%	1.8584	0.0002±0.0003	0.0004±0.0002	21.79
	ABDH [39]	40.62	0.9762	217.12	95.73%	1.8292	0.0003±0.0006	0.0004±0.0003	21.79
	CHAT-GAN	43.17	0.9880	27.16	97.46%	1.8984	0.0025±0.0003	0.0012±0.0015	21.81
3 bpp	HiDDeN† [37]	44.87	0.9947	310.80	50.00%	0.0000	0.0034±0.0019	0.0105±0.0046	21.70
	SteganoGAN [38]	36.45	0.9650	2592.68	91.35%	2.4810	0.0000±0.0000	0.0000±0.0000	21.31
	ABDH [39]	37.95	0.9566	648.31	94.89%	2.6934	0.0000±0.0000	0.0000±0.0000	21.75
	CHAT-GAN	41.84	0.9832	47.72	96.18%	2.7708	0.0015±0.0003	0.0010±0.0003	21.85

†For the embedding capacity 2 bpp and 3 bpp, HiDDeN has 50% message extracting accuracy and 0 effective capacity, and thus it cannot work normally.



(a) cover (b) HiDDeN (c) SteganoGAN (d) ABDH (e) CHAT-GAN (f) HiDDeN (g) SteganoGAN (h) ABDH (i) CHAT-GAN

Fig. 4. Cover and corresponding stego images with 1 bpp payload generated by our method and other steganographic models based on GAN. (a) covers. (b-e) stegos generated by HiDDeN [37], SteganoGAN [38], ABDH [39], and our CHAT-GAN. (f-i) the per-pixel difference between covers and stegos generated by HiDDeN [37], SteganoGAN [38], ABDH [39], and our CHAT-GAN (magnified 5 times).

layer to predict a high-dimensional message vector whereas CHAT-GAN uses convolutional layers. At higher payloads, we observed that HiDDeN failed to converge on message distortion loss. Its extraction error rate is around 50%, which is equivalent to random guessing. Since the error rate exceeds the upper limit for tolerable errors of error-correcting codes, the effective capacity is unexpectedly 0. This does not satisfy the fundamental requirement of steganography, thus HiDDeN cannot work normally at 2 bpp and 3 bpp. The proposed CHAT-GAN outperforms SteganoGAN and ABDH. SteganoGAN and ABDH just use regular CNN as the discriminator and fail to optimize their networks from the perspective of human eye perception. They do not take account of the proper payload distribution in feature maps. In contrast, our improvement on undetectability derives from the XuNet-based discriminator. The optimization through a pre-trained VGG in our method helps to decrease the perceptual loss. The proposed channel-wise attention strategy ensures that the payload is concentrated mainly in more useful channel-wise features, which improves the image quality and message recovery accuracy. From Fig. 4, we can see that stego images generated by our model have the most subtle difference from covers, which intuitively explains the stronger imperceptibility of CHAT-GAN.

It can be noticed that the IS is close for all models. We measured the IS of covers and contrasted it with that of stegos to determine whether or not the generated images were realistic enough. According to our measurement, the IS of covers was 21.85. The value is approximate to that of stegos in all algorithms, so the qualities of these generated images are all acceptable.

2) Comparisons With Distortion Minimization Frameworks: The excellent security of the framework combining distortion function design and STC encoding [8] that minimizes total distortion has been widely proven in classic adaptive steganography. We also compared the proposed method with three typical distortion definition algorithms for gray-scale images, HILL [12], MiPOD [13], and UT-6HF-GAN [34] to embed 1 bpp on COCO dataset. We integrated ACMP [20] into HILL and MiPOD to distribute payloads in color images. For UT-6HF-GAN, we split the 20,000 color images used to train our network into 60,000 gray-scale images as its training set, and allocated payloads evenly across the three channels of color images in the test set to assess the trained model. All embedding algorithms are implemented by the STC simulator with random embedding keys. In previous experiments, to evaluate the security of a certain model $\mathcal{M}$,

TABLE IX
COMPARISONS BETWEEN CHAT-GAN AND TRADITIONAL DISTORTION MINIMIZATION FRAMEWORKS ON COCO DATASET

Criteria		HILL-ACMP [12]	MiPOD-ACMP [13]	UT-6HPF-GAN [34]	CHAT-GAN	
					Scenario 1	Scenario 2
Image Quality	PSNR	58.76	59.17	59.55	46.42	46.42
	SSIM	0.9998	0.9998	0.9998	0.9943	0.9943
	$\mathcal{L}_p$	8.73	8.07	8.58	14.82	14.82
Steganographic Security	Detection Error #1	0.2869±0.0025	0.2743±0.0042	0.2172±0.0056	0.0575±0.0154	0.4215±0.0314
	Detection Error #2	0.1908±0.0291	0.1800±0.0315	0.1668±0.0198	0.0727±0.0076	0.4924±0.0041

we first trained steganalytic classifiers on sufficient pairs of covers and stegos generated by $\mathcal{M}$, and then used the classifiers to detect cover-stego pairs still produced by $\mathcal{M}$. The key point here is that we made the assumption that the steganalyst had access to the exact model $\mathcal{M}$ so the data sets for training and testing steganalyzers were all generated by it. In this case, the detector can easily overfit the embedding model and tend to show a relatively low error rate on the test set. In fact, the parameters of the exact model $\mathcal{M}$ should be kept as a private key by steganographers, and the previous experiments can be regarded as a simulation of the situation where the private key is exposed.

Traditional methods [12], [13], [34] always use random keys to create stegos for training the detector, which presumes that the private key cannot be obtained by a steganalyst. For fair comparisons, we considered the case where the steganographer's model was inaccessible to the steganalyst. To simulate the steganalyst, we trained a new model $\mathcal{M}'$ to create stegos for training classifiers and tested the classifiers on datasets produced by the previous model $\mathcal{M}$. Specifically, there are two scenarios considered: (1) Scenario 1: The seed for randomly initializing the model is unknown. $\mathcal{M}'$ was initialized by a different seed. (2) Scenario 2: Some hyperparameters are unknown. Here we trained $\mathcal{M}'$ using a different truncation factor. It is undeniable that these two scenarios are indeed possible according to Kerckhoffs's principle.

Table IX shows the comparative results with traditional steganographic schemes HILL-ACMP [12], MiPOD-ACMP [13], and UT-6HPF-GAN [34] for embedding 1 bpp payload on COCO dataset. Although CHAT-GAN is inferior in image quality (such as PSNR, SSIM, $\mathcal{L}_p$), it could acquire stronger steganographic security when the hyperparameters are kept secret. Traditional algorithms achieve better image quality owning to their distortion minimization frameworks and ± 1 embedding strategy. We use a network to generate stego images directly and a CNN model is trained to learn the difference between cover images and stego images. However, the stego images created by CHAT-GAN are visually indistinguishable. Note that the steganographic security is more significant in the research of steganography. CHAT-GAN is more flexible because various factors can be decided as secret keys by steganographers. In Scenario 2, a hyperparameter is selected as the secret key. Different hyperparameters will cause obvious changes in embedding patterns, forcing the detector to make more misjudgments. The results given in Table IX have shown that CHAT-GAN has stronger undetectability than HILL-ACMP [12], MiPOD-ACMP [13], and UT-6HPF-GAN [34]. We can further get the enlightenment that steganographers must keep model weights secret in practical applications, and better security can be obtained by keeping hyperparameters secret.

G. Generalization to the Different Data Set

We considered a situation where a steganographer trained models on a source of images but used images from a different source as covers in practice. To illustrate the generalization of CHAT-GAN to new data, we also evaluated it on 1,000 images from the standard color BOSSBase image set [60]. We first applied Photoshop CS6 for demosaicking the raw images in BOSSBase, and then resampled the obtained images to true color images with the size of 256×256 using a bilinear kernel.

The generalization performance on the BOSSBase is also evaluated for HiDDeN, SteganoGAN, and ABDH. At this stage, we directly used the previously trained steganalyzers when detecting each model with a given embedding rate. The results in Table X compares CHAT-GAN with [37]–[39] for embedding various payloads on BOSSBase. The table demonstrates that these approaches all generalize well to other image sets on the whole, which may be the common advantage of this category of steganographic methods. By contrast, the performance of CHAT-GAN is better than that of SteganoGAN and ABDH. For the embedding capacity 1 bpp, CHAT-GAN outperforms HiDDeN. With the increase of the embedding capacity, HiDDeN still suffers from high extraction error rates, which would be unuseful for steganography. The IS of cover images we measured was 10.32. It is close to all the stego images whatever the algorithm and capacity are, which again verifies the acceptability of the visual quality of these images.

It is worth noting that the detection errors of these methods on BOSSBase have increased compared with their detection errors on COCO. This is because we did not train steganalyzers on the pairs of covers and stegos from BOSSBase. This simulation is closer to real communication environments where the steganalyst can no longer train the detector on the image set used by the steganographers.

H. Discussions

1) Performance on Different Message Distributions: We trained our models on random messages whose bits obey an independent and identical Bernoulli distribution with $p = 0.5$, where p represents the probability that the bit value is 1. To investigate the generalization of the model for other

TABLE X
COMPARISONS BETWEEN CHAT-GAN AND OTHER GAN-BASED STEGANOGRAPHIC ALGORITHMS WITH VARIOUS PAYLOADS ON BOSSBASE

Capacity	Method	PSNR	SSIM	$\mathcal{L}_p$	Accuracy	Effective Capacity	Detection Error #1	Detection Error #2	IS
1 bpp	HiDDeN [37]	42.89	0.9808	514.15	58.24%	0.1648	0.0617±0.0050	0.0255±0.0025	10.30
	SteganoGAN [38]	43.08	0.9825	225.23	98.81%	0.9762	0.0376±0.0132	0.0489±0.0099	10.35
	ABDH [39]	43.98	0.9829	174.77	97.77%	0.9554	0.0417±0.0510	0.0167±0.0087	10.32
	CHAT-GAN	47.27	0.9920	17.20	99.36%	0.9872	0.0890±0.0365	0.0787±0.0452	10.33
2 bpp	HiDDeN† [37]	49.47	0.9956	168.80	50.00%	0.0000	0.1785±0.0141	0.1865±0.0631	10.31
	SteganoGAN [38]	40.07	0.9678	399.48	96.98%	1.4792	0.0412±0.0098	0.0375±0.0198	10.36
	ABDH [39]	40.78	0.9633	250.96	95.91%	1.8364	0.0247±0.0314	0.0100±0.0078	10.30
	CHAT-GAN	43.56	0.9817	29.71	98.13%	1.9252	0.0511±0.0204	0.0455±0.0332	10.31
3 bpp	HiDDeN† [37]	45.83	0.9925	278.99	49.99%	0.0000	0.1490±0.0115	0.1049±0.0334	10.35
	SteganoGAN [38]	37.98	0.9536	1587.43	93.14%	2.5884	0.0194±0.0086	0.0261±0.0022	10.24
	ABDH [39]	38.49	0.9394	772.64	94.68%	2.6808	0.0137±0.0149	0.0020±0.0006	10.30
	CHAT-GAN	42.22	0.9745	49.85	97.18%	2.8308	0.0323±0.0202	0.0266±0.0174	10.31

†For the embedding capacity 2 bpp and 3 bpp, HiDDeN has 50% message extracting accuracy and 0 effective capacity, and thus it cannot work normally.

TABLE XI
PERFORMANCE OF CHAT-GAN ON DIFFERENT MESSAGE DISTRIBUTIONS

Criteria	$p = 0.00$	$p = 0.25$	$p = 0.75$	$p = 1.00$	Non-IID
PSNR	48.58	46.03	46.47	48.97	46.43
SSIM	0.9963	0.9937	0.9943	0.9965	0.9943
$\mathcal{L}_p$	12.39	16.37	15.29	11.11	14.81
Accuracy	41.32%	96.43%	96.55%	60.19%	99.07%

message distributions, we changed the probability p in messages for testing. An example of non-IID was also tried, where each bit obeyed a Bernoulli distribution with a different probability. As illustrated in Table XI, when p approaches 0 and 1, the image quality increases but the message extraction accuracy reduces. The best performance is observed for messages with non-IID. This is because the message pattern with the distribution of p closer to 0.5 or the non-IID is similar to that in training. Since the performance attenuates when p deviates from 0.5 to both sides, training on the Bernoulli distribution with $p = 0.5$ is the best choice. The binary sequence transformed from actual text information also approximately follows this distribution.

2) Efficiency of the Proposed Method: We compare the proposed CHAT-GAN with the networks in [37]–[39] on computational efficiency by calculating the numbers of floating point operations (FLOPs) and parameters (#Param). These two metrics are influenced by the input image size and capacity. For fair comparisons, CHAT-GAN and other CNN-based steganographic algorithms [37]–[39] are all used to embed 1 bpp payload into a 16×16 image for calculating the metrics. Note that we ignore the discriminators in each method and focus only on the complexity of the generator and extractor. For a convolutional layer, suppose the kernel size is $k_w \times k_h$, the numbers of input channels and output channels are C_{in} and C_{out}, the size of the output feature map is $f_w \times f_h$. Then the FLOPs (including the computation of multiplication, addition, and bias) is $2 \times (k_w \times k_h \times C_{in}) \times f_w \times f_h \times C_{out}$ and the #Param (including weights and bias) is $C_{out} \times (k_w \times k_h \times C_{in} + 1)$. For a fully-connected layer, suppose the numbers of input features and output features are f_{in} and f_{out}, then the FLOPs is $2 \times f_{in} \times f_{out}$ and the #Param is $f_{out} \times (f_{in} + 1)$.

TABLE XII
COMPARISONS OF COMPUTATIONAL COMPLEXITY BETWEEN CHAT-GAN AND OTHER GAN-BASED STEGANOGRAPHIC ALGORITHMS

Criteria	HiDDeN [37]	SteganoGAN [38]	ABDH [39]	CHAT-GAN
FLOPs ($\times 10^6$)	172.64	15.83	6047.91	15.89
#Param ($\times 10^3$)	737.80	61.66	24944.86	62.43

According to the above calculation rule, we obtained the FLOPs and #Param of CHAT-GAN and other CNN-based embedding methods. The results in Table XII show that our model has relatively low computational complexity and storage overhead. The channel attention module in CHAT-GAN contains two fully-connected layers with 8480 FLOPs and 128 parameters. Both generator and extractor in CHAT-GAN have four convolutional layers and three channel attention modules. The FLOPs of the convolutional layers in the generator are 253952, 2465792, 4825088, 671232, and the #Param are 960, 9600, 18816, 2622. The FLOPs of the convolutional layers in the extractor are 253952, 2392064, 4751360, 221440, and the #Param are 960, 9312, 18528, 865. Hence, the total FLOPs of CHAT-GAN is 15.89 M and #Param is 62.43 K. HiDDeN encodes the message with 6 convolutional layers and decodes it with 8 convolutional layers and one fully-connected layer, which brings expensive computational cost. SteganoGAN has 8 convolutional layers totally. Despite its low cost, it is inferior in steganography performance, as shown in Table VIII and Table X. The primary cause for the high complexity of ABDH is the spatial attention module. It is the feature extractor backbone of ResNet50 [61], which has 6.02 billion FLOPs and 23.51 million parameters. By contrast, the channel attention module in CHAT-GAN is extremely lightweight.

For traditional algorithms [12], [13], [34], we compared the average embedding time of each image on the test set. HILL-

ACMP [12] and MiPOD-ACMP [13] run on CPU and the average embedding time is 0.7815 s and 2.6706 s per image. UT-6HPF-GAN [34] and our network run on GPU and it averagely takes 0.0136 s and 0.0064 s for each method to generate a stego image. Thus, the proposed algorithm always runs faster.

IV. Conclusion

In this paper, we present a novel GAN-based end-to-end network architecture for image steganography with channel attention taken into consideration. Inside the model, the generator hides the secret information within covers and generates stegos while the extractor recovers messages from stego images. We design a channel attention module to take full advantage of channel relationships. It can be easily incorporated into the networks to better modify the channel features. In terms of the design of the discriminator, we borrow an advanced CNN for steganalysis and adjust the channel and the number of high-pass filters in its pre-processing layer to detect color images. We find that the model complexity and performance of the discriminator will affect the power of the generator to some extent. When putting the trained model into practice, we can employ an error-correcting algorithm to guarantee the perfect recovery of secret information. In the circumstance of using a kind of common error correction code, we offer analysis on the calculation of effective capacity. Experiments show that our model outperforms existing GAN-based methods for generating stegos. The steganographic security is preferable to traditional algorithms when the model hyperparameters are unknown to the steganalyst. We also verify the generalization of our model to another image set, on which CHAT-GAN is still superior to other models.

In future work, we hope to continue to explore the property of the methods of generating stego images directly using GAN. With the utilization of deep learning in the study of error correction algorithms, further research for joint training of the error-correcting neural network and the steganographic embedding network is worth considering.

References

[1] B. Mei, Y. Xiao, R. Li, H. Li, X. Cheng, and Y. Sun, "Image and attribute based convolutional neural network inference attacks in social networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 2, pp. 869–879, Apr.–Jun. 2020.

[2] J. Mao, Y. Yang, and T. Zhang, "Empirical analysis of attribute inference techniques in online social network," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 881–893, Apr.–Jun. 2021.

[3] R. J. Anderson and F. A. P. Petitcolas, "On the limits of steganography," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 474–481, May 1998.

[4] Z. Xia, L. Wang, J. Tang, N. Xiong, and J. Weng, "A privacy-preserving image retrieval scheme using secure local binary pattern in cloud computing," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 318–330, Jan.–Mar. 2021.

[5] H. Yang, H. He, W. Zhang, and X. Cao, "FedSteg: A federated transfer learning framework for secure image steganalysis," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1084–1094, Apr.–Jun. 2021.

[6] J. Mielikainen, "LSB matching revisited," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 285–287, May 2006.

[7] S. Dumitrescu, X. Wu, and Z. Wang, "Detection of LSB steganography via sample pair analysis," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1995–2007, Jul. 2003.

[8] T. Filler, J. Judas, and J. Fridrich, "Minimizing additive distortion in steganography using syndrome-trellis codes," *IEEE Trans. Inf. Forensics Secur.*, vol. 6, no. 3, pp. 920–935, Sep. 2011.

[9] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. Int. Workshop Inf. Hiding*, 2010, pp. 161–177.

[10] V. Holub and J. Fridrich, "Designing steganographic distortion using directional filters," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2012, pp. 234–239.

[11] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP J. Inf. Secur.*, vol. 2014, no. 1, pp. 1–13, 2014.

[12] B. Li, M. Wang, J. Huang, and X. Li, "A new cost function for spatial image steganography," in *Proc. IEEE Int. Conf. Image Process.*, 2014, pp. 4206–4210.

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

[14] L. Guo, J. Ni, and Q. S. Yun, "Uniform embedding for efficient JPEG steganography," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 5, pp. 814–825, May 2014.

[15] C. Wang and J. Ni, "An efficient JPEG steganographic scheme based on the block entropy of DCT coefficients," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2012, pp. 1785–1788.

[16] B. Li, M. Wang, X. Li, S. Tan, and J. Huang, "A strategy of clustering modification directions in spatial image steganography," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 9, pp. 1905–1917, Sep. 2015.

[17] T. Denemark and J. Fridrich, "Improving steganographic security by synchronizing the selection channel," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, 2015, pp. 5–14.

[18] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Trans. Inf. Forensics Secur.*, vol. 5, no. 4, pp. 705–720, Dec. 2010.

[19] W. Tang, B. Li, W. Luo, and J. Huang, "Clustering steganographic modification directions for color components," *IEEE Signal Process. Lett.*, vol. 23, no. 2, pp. 197–201, Feb. 2016.

[20] X. Liao, Y. Yu, B. Li, Z. Li, and Z. Qin, "A new payload partition strategy in color image steganography," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 3, pp. 685–696, Mar. 2020.

[21] Y. Wang, W. Zhang, W. Li, X. Yu, and N. Yu, "Non-additive cost functions for color image steganography based on inter-channel correlations and differences," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 2081–2095, 2020.

[22] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 708–712, May 2016.

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

[24] M. Yedroudj, F. Comby, and M. Chaumont, "Yedroudj-Net: An efficient CNN for spatial steganalysis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 2092–2096.

[25] J. Zeng, S. Tan, B. Li, and J. Huang, "Large-scale JPEG image steganalysis using hybrid deep-learning framework," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 5, pp. 1200–1214, May 2018.

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

[27] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[28] 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.

[29] X. Chen, Z. Zhang, A. Qiu, Z. Xia, and N. Xiong, "A novel coverless steganography method based on image selection and StarGAN," *IEEE Trans. Netw. Sci. Eng.*, to be published, doi: 10.1109/TNSE.2020.3041529.

[30] D. Volkhonskiy, I. Nazarov, and E. Burnaev, "Steganographic generative adversarial networks," in *Proc. Int. Conf. Mach. Vis.*, 2020, pp. 991–1005.

[31] H. Shi, J. Dong, W. Wang, Y. Qian, and X. Zhang, "SSGAN: Secure steganography based on generative adversarial networks," in *Proc. Pacific Rim Conf. Multimedia*, 2017, pp. 534–544.

[32] H. Zi, Q. Zhang, J. Yang, and X. Kang, "Steganography with convincing normal image from a joint generative adversarial framework," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2018, pp. 526–532.

[33] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Signal Process. Lett.*, vol. 24, no. 10, pp. 1547–1551, Oct. 2017.

[34] J. Yang, D. Ruan, J. Huang, X. Kang, and Y. Shi, "An embedding cost learning framework using GAN," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 839–851, 2020.
[35] J. Yang, D. Ruan, X. Kang, and Y.-Q. Shi, "Towards automatic embedding cost learning for JPEG steganography," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, 2019, pp. 37–46.
[36] J. Hayes and G. Danezis, "Generating steganographic images via adversarial training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1954–1963.
[37] J. Zhu, R. Kaplan, J. Johnson, and F. Li, "HiDDeN: Hiding data with deep networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 657–672.
[38] K. A. Zhang, A. Cuestainfante, L. Xu, and K. Veeramachaneni, "SteganoGAN: High capacity image steganography with GANs," 2019, *arXiv:1901.03892*.
[39] C. Yu, "Attention based data hiding with generative adversarial networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 1120–1128.
[40] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 2, pp. 300–304, 1960.
[41] G. Huang, Z. Liu, L. V. Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.
[42] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 3, pp. 868–882, Jun. 2012.
[43] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.
[44] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.
[45] J. Johnson, A. Alahi, and F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.
[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.
[47] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.
[48] A. Almohammad and G. Ghinea, "Stego image quality and the reliability of PSNR," in *Proc. Int. Conf. Image Process. Theory, Tools Appl.*, 2010, pp. 215–220.
[49] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[50] M. Goljan, J. Fridrich, and R. Cogranne, "Rich model for steganalysis of color images," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, 2014, pp. 185–190.
[51] J. Kodovsky, J. Fridrich, and V. Holub, "Ensemble classifiers for steganalysis of digital media," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 2, pp. 432–444, Apr. 2012.
[52] J. Zeng, S. Tan, G. Liu, B. Li, and J. Huang, "WISERNet: Wider separate-then-reunion network for steganalysis of color images," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 10, pp. 2735–2748, Oct. 2019.
[53] T. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[55] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3024–3033.
[56] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11531–11539.
[57] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.
[58] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2223–2232.
[59] T. Salimans *et al.*, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 2234–2242.
[60] P. Bas, T. Filler, and T. Pevný, "'Break our steganographic system': The ins and outs of organizing BOSS," in *Proc. Int. Workshop Inf. Hiding*, 2011, pp. 59–70.
[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

Jingxuan Tan received the B.E. degree in information security from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2021. He is currently working toward the M.S. degree with the School of Computer Science, Fudan University, Shanghai, China. His research interests include image steganography and digital watermarking.

Xin Liao (Senior Member, IEEE) received the B.E. and Ph.D. degree in information security from the Beijing University of Posts and Telecommunications, Beijing, China, in 2007 and 2012, respectively. He is currently an Associate Professor and a Doctoral Supervisor with Hunan University, Changsha, China. He was a Postdoctoral Fellow with the Institute of Software, Chinese Academy of Sciences, Beijing, China, and also a Research Associate with The University of Hong Kong, Hong Kong. From 2016 to 2017, he was a Visiting Scholar with the University of Maryland, College Park, MD, USA. His current research interests include multimedia forensics, steganography, and watermarking. He is an Associate Editor for the *IEEE Signal Processing Magazine*. He is also a Member of Technical Committee (TC) on Multimedia Security and Forensics of Asia Pacific Signal and Information Processing Association, TC on Computer Forensics of Chinese Institute of Electronics, and TC on Digital Forensics and Security of China Society of Image and Graphics.

Jiate Liu received the B.E. degree in information Security from Hunan University, Changsha, China, in 2019. He is currently working toward the M.E. degree in computer science with Tsinghua University, Beijing, China. His research interests include graph networks and reinforcement learning.

Yun Cao (Member, IEEE) received the B.S. degree from Tsinghua University, Beijing, China, in 2006, and the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2012. He is currently an Associate Professor with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, and also with the School of Cyber Security, University of Chinese Academy of Sciences. His research interests include information hiding, digital forensics, and video processing.

Hongbo Jiang (Senior Member, IEEE) received the Ph.D. degree from Case Western Reserve University, Cleveland, OH, USA, in 2008. He is currently a Full Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. He was a Professor with the Huazhong University of Science and Technology, Wuhan, China. His research interests include computer networking, especially algorithms and protocols for wireless and mobile networks. He is the Editor of the IEEE/ACM TRANSACTIONS ON NETWORKING, an Associate Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING, and an Associate Technical Editor for the *IEEE Communications Magazine*.