RESEARCH-ARTICLE

Color Image Steganography Using Generative Adversarial Networks with a Phased Training Strategy

SAIXING ZHOU, Sun Yat-Sen University, Guangzhou, Guangdong, China

MIAOXIN YE, Sun Yat-Sen University, Guangzhou, Guangdong, China

WEIQI LUO, Sun Yat-Sen University, Guangzhou, Guangdong, China

XIN LIAO, Hunan University, Changsha, Hunan, China

KANGKANG WEI, Nanchang University, Nanchang, Jiangxi, China

Published: 18 June 2025

IH&MMSEC '25: ACM Workshop on Information Hiding and Multimedia Security
June 18 - 20, 2025
San Jose, USA

Conference Sponsors:
SIGMM

Color Image Steganography Using Generative Adversarial Networks with a Phased Training Strategy

Saixing Zhou
Sun Yat-sen University
Guangzhou, China
zhousx7@mail2.sysu.edu.cn

Miaoxin Ye
Sun Yat-sen University
Guangzhou, China
yemx8@mail2.sysu.edu.cn

Weiqi Luo*
Sun Yat-sen University
Guangzhou, China
luoweiqi@mail.sysu.edu.cn

Xin Liao
Hunan University
Changsha, China
xinliao@hnu.edu.cn

Kangkang Wei
Nanchang University
Nanchang, China
kkwei@ncu.edu.cn

Abstract

Most existing steganographic techniques are primarily designed for grayscale images. When directly applied to color images without considering inter-channel color interactions, their security can be significantly compromised. In this paper, we propose a novel color image steganography method based on generative adversarial networks (GANs). Our framework features a dual-branch generator and a discriminator equipped with multiple steganalytic networks, each focused on a specific color channel. This architecture enables the progressive learning of asymmetric embedding costs across channels from scratch. We also introduce a phased training strategy that facilitates the learning of inter-channel color interactions and optimizes the security of each color channel in distinct phases, improving overall security. Furthermore, we propose a new adaptive update strategy and introduce the Mean Absolute Deviation (MAD) loss function to maintain a dynamic balance between the generator and discriminator, thereby progressively enhancing the generator's steganographic performance during training. Extensive comparative experiments demonstrate that the proposed method achieves state-of-the-art security against color image steganalysis. Comprehensive ablation studies further validate the effectiveness and rationale behind our approach.

Keywords

Image steganography, Generative adversarial networks (GANs), Embedding costs

ACM Reference Format:
Saixing Zhou, Miaoxin Ye, Weiqi Luo, Xin Liao, and Kangkang Wei. 2025. Color Image Steganography Using Generative Adversarial Networks with a Phased Training Strategy. In *ACM Workshop on Information Hiding and Multimedia Security (IH&MMSEC '25), June 18–20, 2025, San Jose, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3733102.3733112

*Corresponding Author.

IH&MMSEC '25, San Jose, CA, USA

ACM ISBN 979-8-4007-1887-8/25/06
https://doi.org/10.1145/3733102.3733112

1 Introduction

Image steganography [17] leverages the statistical redundancy of cover images and the perceptual characteristics of human vision to embed secret information by altering specific embedding units (e.g., pixel values, DCT coefficients), facilitating covert communication. Current steganography techniques are based on the minimal distortion framework [7], where the focus is on designing distortion functions for each embedding unit, followed by encoding schemes (e.g., STC [6], SPC[26]) for embedding secret information. Typical method includes HUGO [18], WOW [9], S-UNIWARD [10], and HILL [13]. These methods allocate lower embedding costs to texture-rich regions and higher costs to smoother, simpler areas. This approach embeds more secret data in regions where pixel changes are less perceptible and harder to model, enhancing the steganography security. The aforementioned methods are symmetric steganography techniques that treat the costs of embedding +1 and −1 pixel modifications equally. In contrast, Li et al. and Denemark et al.proposed asymmetric steganography strategy, CMD[14] and SMD[4], respectively. These approaches segment the cover image into non-overlapping sub-images, sequentially embed data, and update the costs in unmodified areas based on the direction of pixel alterations. This results in asymmetric embedding costs, enhancing the overall security of the steganography.

However, all of the methods mentioned above are designed for grayscale images. Directly applying these techniques to the individual color channels of color images may result in noticeable embedding artifacts, thereby reducing their security. Although some steganographic methods have been developed for color images, such as Tang et al.'s CMD-C [22] strategy, which extends the CMD method to color images, it is less effective due to its failure to account for the correlations between the color channels. Liao et al. proposed the ACMP [16] strategy, which aggregates the changes in the embedded message by adjusting the initial distortion at modification points across all three channels, concentrating the changes in texture regions. Wang et al. introduced the asymmetric GINA [23] strategy, inspired by CMD, that considers the correlations and differences between channels. This approach ensures synchronization between the red (r) and blue (b) channels with the green (g) channel and selectively updates embedding costs based on the complexity priority principle, significantly enhancing the security of color image steganography. Please note that the methods mentioned above primarily rely on existing grayscale algorithms (e.g.,

S-UNIWARD and HILL) to initialize the embedding costs for each channel before applying the proposed cost update strategies. With the rise of deep learning, adversarial examples and Generative Adversarial Networks (GANs) have also been applied to color image steganography. For example, Qin et al. introduced the adversarial sample-based method GEAP [19], which utilizes the existing color steganography - CPV [20] to initialize the embedding costs and then updates it through an adversarial mechanism to resist detection by target steganalysis models. However, its security is found to be weaker compared to the GINA strategy.

Unlike the methods mentioned above, Liao et al. proposed the first GAN-based approach, CIS-GAN [15], for designing the steganographic distortion function for color images. This method can learn the asymmetric embedding costs of color images from scratch. However, it has two limitations. First, the model uses a relatively weaker steganalyzer, Xu-Net [27], as its discriminator, which likely limits the performance of the generator. Second, the model adopts a simultaneous embedding strategy, embedding information into all color channels at the same time. This approach makes it difficult to effectively capture the correlations between the channels, limiting its ability to enhance the security of the steganographic method. As a result, its security performance is unsatisfactory.

In this paper, we propose a novel color image steganography method based on generative adversarial networks (GANs). The method employs a phased embedding strategy that takes into account the correlations between color channels, dividing the training process of the network model into three stages. Each stage learns the embedding costs for a specific color channel. In terms of model design, each stage is equipped with a dual-branch generator and a discriminator composed of multiple steganalysis networks. Additionally, we introduce a simple yet effective mean absolute deviation (MAD) loss, which dynamically adjusts the image discrimination loss weights of all networks in the discriminator. This helps balance the training of the generator and discriminator, while gradually improving the steganographic performance of the generator. Our method enables the asymmetric embedding costs for each color channel to be learned step by step from scratch, without relying on traditional steganography algorithms. Extensive experiments demonstrate that our approach outperforms existing methods in terms of security performance. In summary, the main contributions of this paper are as follows:

- We propose a novel GAN-based framework for color image steganography that learns the asymmetric embedding costs of different color channels from scratch. This framework employs a phased embedding strategy to effectively capture inter-channel correlations and systematically explores the influence of different channel embedding orders on steganographic security, ultimately improving the security.
- We introduce the MAD loss function, which dynamically adjusts the loss weights of different networks in discriminator. This mechanism ensures a balanced training process between the generator and discriminator, stabilizes model convergence, and progressively improves the generator's steganographic performance.
- Extensive experiments on color image datasets, including ALASKA II and BOSSBase, demonstrate that our method significantly outperforms existing approaches in terms of security. Furthermore, many ablation studies validate the effectiveness of our network design and loss function in improving steganographic performance.

2 Proposed Method

In this section, we give a detailed introduction to the proposed GANs model, covering the phased training and embedding process, the design of the generators and discriminators, as well as the adaptive update strategy and loss functions.

2.1 Phased training-embedding process

As shown in Fig. 1, the proposed method consists of three stages, with each stage dedicated to training and embedding for a specific color channel. The training and embedding process of Stage 1 is illustrated in Fig. 1-(a), while the subsequent stages (as shown in Fig. 1-(b) and Fig. 1-(c)) follow a similar approach. The key difference is that the input for later stages incorporates the resulting images embedded in the preceding stages. Based on our comparative experiments in Section 3-3.3, the preferred order for training and embedding the color channels is $r \rightarrow b \rightarrow g$.

For clarity, we define the following symbols. Assume that the color cover images C have a size of $H \times W \times 3$ and are in the RGB (red, green, blue) color space. The individual channels are denoted as C_r, C_g, C_b, each with a size of $H \times W \times 1$. During each stage, the resulting stego images for the three color channels from the embedding simulator in the training process are denoted as S'_r, S'_g, S'_b. In the embedding process, the resulting stego images after applying the STC are denoted as S_r, S_g, S_b. Since the training and embedding processes are similar across all stages, we will focus on providing a detailed description of the process in the first stage.

In Stage 1, we first train a generator G_r to learn an asymmetric embedding probability for the red channel of the cover image C. Specifically, we input the cover image C_r into the generator G_r, which outputs two probability maps for the red channel, denoted as $P^{+1} = (p^{+1}_{i,j})^{H\times W}$ and $P^{-1} = (p^{-1}_{i,j})^{H\times W}$, respectively. In these maps, $p^{+1}_{i,j}$ ($p^{-1}_{i,j}$) represents the probability that the pixel value at position (i, j) is increased by +1 (−1) due to the embedding of secret information. We then use an embedding simulator to simulate the data embedding. To achieve this, we firstly generate a random noise $N = (n_{i,j})^{H\times W}$, where $n_{i,j} \in [0, 1]$. By comparing the P and N , we can obtain the r-channel modification map $M = (m_{i,j})^{H\times W}$ by the embedding simulator as follows:

$$m_{i,j} = \begin{cases} +1, & n_{i,j} < p^{+1}_{i,j} \\ -1, & n_{i,j} > 1 - p^{-1}_{i,j}, \\ 0, & otherwise. \end{cases} \tag{1}$$

We get the simulated r-channel stego image by $S^{'}_r = C_r + M$. Following this, both C_r and $S^{'}_r$ serve as inputs for the discriminator D_r, which outputs its discriminant probability. This output is then backpropagated to the generator G_r. After iterative training, G_r learns the asymmetric embedding probabilities of C_r, ensuring that the embedding modifications occur predominantly in texture-rich areas that are more challenging to detect.

Once the training process is complete, we use the trained generator G_r to determine the probability maps P for C_r, which are then

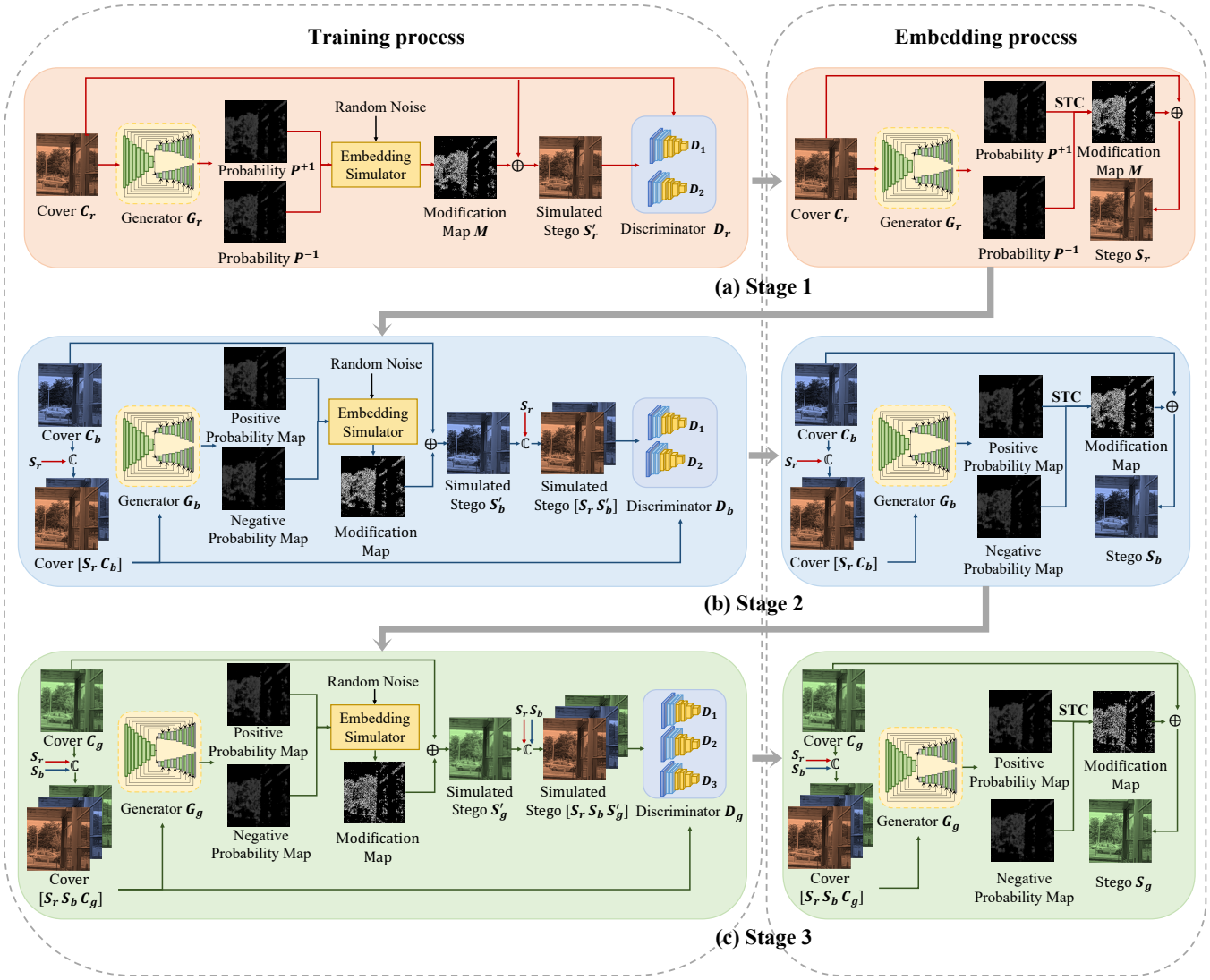


Figure 1: The phased training and embedding process of the proposed model. $\mathbb{C}$ represents the operation of concatenating images along the channels. $\oplus$ represents the operation of element-wise addition.

converted into asymmetric embedding costs ρ for C_r as follows:

$$\begin{cases} \rho_{i,j}^{+1} = \ln\left(\frac{1}{p_{i,j}^{+1}} - 2\right), \\ \rho_{i,j}^{-1} = \ln\left(\frac{1}{p_{i,j}^{-1}} - 2\right), \\ \rho_{i,j}^{0} = 0. \end{cases} \tag{2}$$

Finally, we obtain the stego image S_r for the red channel of the color cover image C by applying the STC to the embedding costs ρ.

In Stage 2, the stego image S_r in stage 1 is concatenated with the cover image C_b along channel dimension to create a two-channel cover image $[S_r\ C_b]$, which is input into the generator G_b. G_b learns the embedding locations and modification directions within C_b by capturing correlations between S_r and C_b. Similar to Stage 1, the simulated b-channel stego image $S_b^{'}$ is generated, and the image pair $[S_r\ C_b]$ and $[S_r\ S_b^{'}]$ is input into discriminator D_b. This process ensures synchronization between the color channels. After training, asymmetric embedding probabilities are converted into embedding costs, and STC is applied to generate the b-channel stego image S_b. In Stage 3, the stego images S_r and S_b obtained in Stages 1 and 2 are concatenated with the cover channel image C_g to serve as the input to the generator G_g. Similar to Stage 2, we ultimately obtain the g-channel stego image S_g.

Finally, after Stage 1, 2, and 3, we obtain the stego images for the three channels of the cover image C.

2.2 Design of Generator

As shown in Fig. 2, the generator for each color channel (i.e., G_r, G_b, G_g in Fig. 1) is based on the U-Net architecture [21], and it aims to generate the corresponding embedding probabilities for a specific color channel. Note that, since we need to concatenate

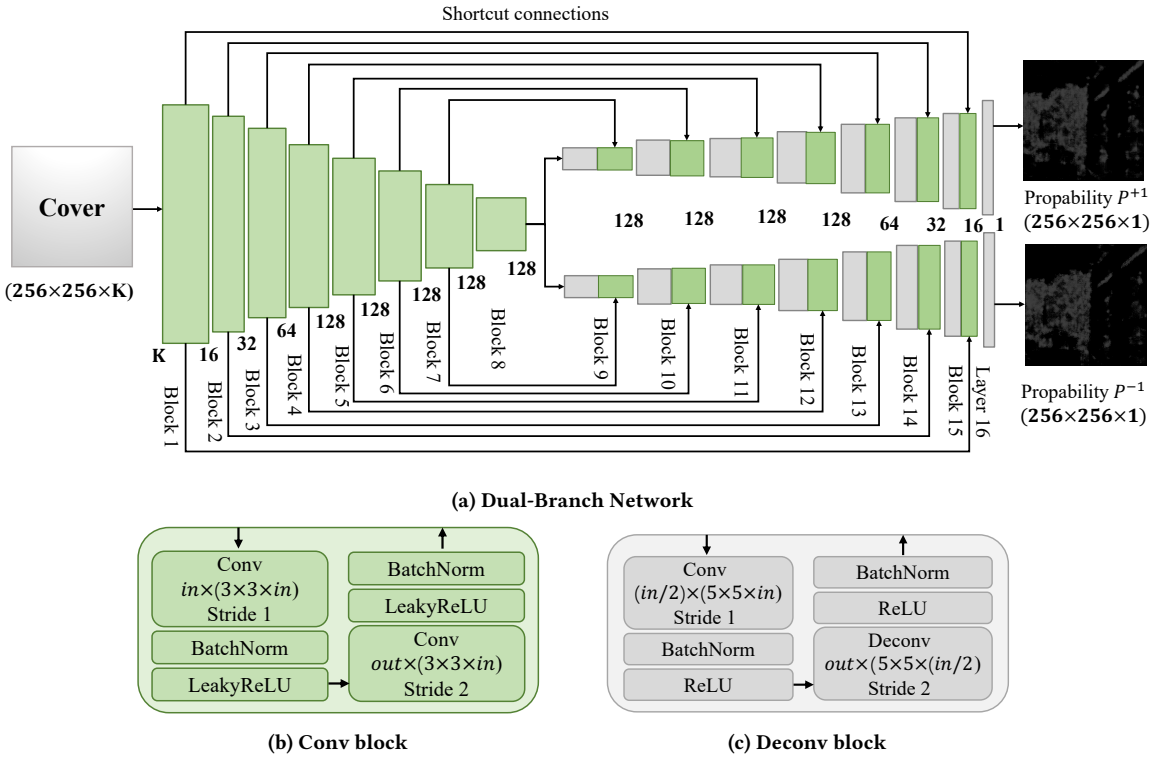


Figure 2: The generator structure of the proposed method. $K = 1, 2, 3$.

the stego images obtained from previous stages, the number of input cover channels increases from 1 to 3 at different stages. To ensure the same generator architecture can effectively process cover images with different number of channels, the number of convolutional kernels in the first layer of Block 1 is adjusted to match the number of channels in the input cover images, as shown in Fig. 2-(a). The downsampling layer consists of 8 convolutional blocks, each containing two convolutional layers, with batch normalization and LeakyReLU applied afterwards, as shown in Fig. 2-(b). The upsampling layer is composed of two branches with identical structures but distinct parameters. Each block consists of a convolutional layer and a deconvolutional layer, with batch normalization and ReLU applied afterwards, as shown in Fig. 2-(c).

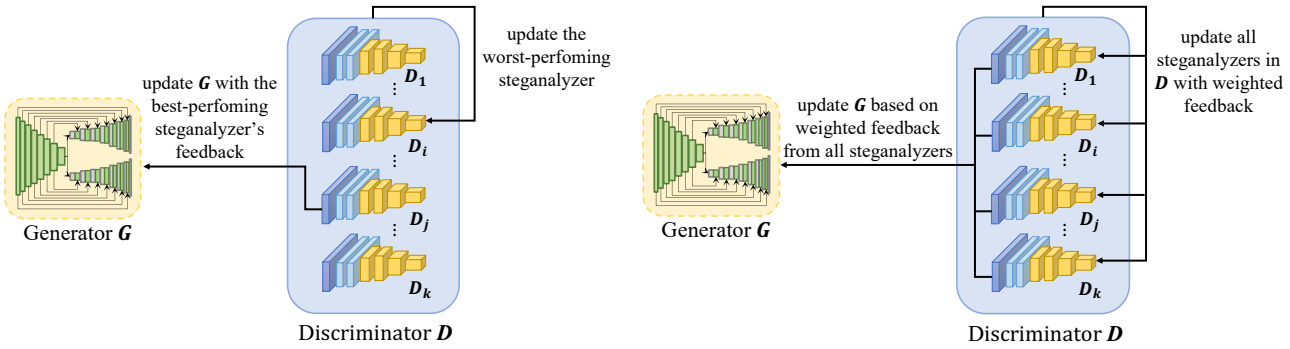
2.3 Design of Discriminator

During the training stage, the proposed discriminators aim to determine whether an input image is a cover image or a stego image, providing valuable feedback to update the network parameters for both the generators and the discriminators. Unlike existing methods like CIS-GAN, which employs a single and relatively weak steganalyzer (Xu-Net) within its discriminator, the proposed approach uses multiple steganalyzers combined with an adaptive update strategy to ensure effective training. Furthermore, the selection of steganalyzers plays a crucial role in determining the final security level. If the discriminator's performance is weak, its inaccurate decision boundary provides limited feedback, which can restrict the performance of the generator. On the other hand, if the discriminator's performance is strong, it may overly penalize the generator, negatively impacting its performance due to excessively harsh feedback, as described in [12]. To address this, we carefully selected steganalyzers from existing image steganalysis methods, including CovNet [5], WISERNet [28], and UCNet [25], to construct the discriminators.

As shown in Fig. 1, the input channels for the three stages are 1, 2, and 3, respectively. Similar to the generator, we made subtle modifications to the steganalytic networks to enable them to process cover and stego images with varying numbers of input channels. Specifically, in the first layer of CovNet, we adjusted the number of convolutional kernels to match the number of input image channels. Additionally, during the preprocessing stages of WISERNet and UCNet, we applied convolution and merging operations for each channel, based on the number of input channels, and modified the first layer's convolutional kernels to align with the merged feature map channels. Based on our experiments (refer to Section 3-3.4), the default architecture for the discriminators D_r and D_b consists of CovNet and WISERNet, while the architecture for discriminator D_g includes CovNet, WISERNet, and UCNet-v1 — a simplified version introduced in UCNet [25] — its detection performance is relatively weaker than UCNet.

2.4 Adaptive Update Strategy

To maintain a dynamic balance between the generators and discriminators during training, we attempted to adopt the update strategy proposed (i.e., variant 1 in Section 3-3.5) by Huang et al. [11]. In



(a) Huang et al.'s strategy[11] **(b) Porposed's strategy**

Figure 3: Comparison of update strategies between Huang et al.'s method [11] and the proposed method.

each iteration, this method optimizes the generator's parameters using feedback from the steganalyzer with the best discrimination performance, while simultaneously updating the parameters of the steganalyzer with the weakest performance, as shown in Fig. 3-(a). However, this strategy has two main limitations. First, relying solely on feedback from the best-performing network overlooks valuable insights from other networks. These networks provide different perspectives, capturing subtle variations in the statistical features of images before and after steganography. Incorporating this diverse feedback could enhance the generator's ability to learn the embedding costs of the images, ultimately leading to more secure stego images. Second, updating only the network with the weakest discrimination performance does not effectively improve the overall discrimination capabilities of the discriminator. This limitation prevents the discriminator from fully capturing the image features, thereby restricting the generator's ability to produce high-quality stego images.

To address these limitations, we adopt the adaptive update strategy shown in Fig. 3-(b). During each iteration, the generator's parameters are updated based on weighted feedback from all steganalyzers in discriminator, with the feedback from the best-performing steganalyzer given the highest weight. Conversely, when optimizing the discriminator, all networks' parameters are updated, but the worst-performing steganalyzer is assigned the highest weight. This strategy is implemented in conjunction with the MAD loss (refer next subsection) we propose. In our method, we use cross-entropy to assess the performance of the i-th steganalyzer D_i within the discriminators. The image discrimination loss is defined as follow:

$$l_{D_i}(X,Y) = -v_0 log(D_i(X)) - v_1 log(D_i(Y)), \tag{3}$$

where $D_i(X)$ and $D_i(Y)$ represent the classification results generated by D_i for cover images X and stego images Y, respectively, while the ground-truth labels for X and Y are denoted as v_0 and v_1. A higher value of l_{D_i} indicates weaker discriminative performance of the steganalyzer in the current iteration.

2.5 Loss Functions

Mean Absolute Deviation Loss: The Mean Absolute Deviation (MAD) loss is as follows:

$$l_{MAD} = \frac{1}{k}\sum_{i=1}^{k}\left| l_{D_i}(X,Y) - \frac{1}{k}\sum_{i'=1}^{k} l_{D_{i'}}(X,Y)\right|, \tag{4}$$

where k denotes the number of steganalyzers in the discriminator, with $k \in \{2,3\}$, and i and i' refer to the i-th and i'-th steganalyzers D_i and $D_{i'}$, respectively. The loss l_{MAD} is used to evaluate the performance disparity among the networks within the discriminator. A larger value of l_{MAD} indicates a greater imbalance in the performance of the steganalyzers, which weakens the overall performance of the discriminator. Therefore, during each iteration, the discriminator aims to minimize this value, while the generator seeks to maximize it.

In the following sections, we will incorporate l_{MAD} into the loss function design for both the discriminator and the generator.

Discriminator Loss: The discriminator loss l_D is defined as follows:

$$l_D = \max_{i\in\{1,\ldots,k\}}\{l_{D_i}(X,Y)\} + \omega_{max}\cdot l_{MAD}, \tag{5}$$

note that l_D consists of two terms: the first term is the discrimination loss from the weakest-performing steganalyzer within the discriminator at each iteration, and the second term is the mean absolute deviation loss, weighted by ω_{max}, which is defined by the following formula:

$$\omega_{max} = e^{(1-\max\limits_{i\in\{1,\ldots,k\}}\{l_{D_i}(X,Y)\})} / \sum_{i'=1}^{k} e^{(1-l_{D_{i'}}(X,Y))}. \tag{6}$$

Generator Loss: The Generator loss l_G is defined as follows:

$$l_G = \alpha\cdot l_G^1 + \beta\cdot l_G^2. \tag{7}$$

Note that l_G comprises two components, namely l_G^1 and l_G^2 . l_G^1 represents the embedding loss of generator that ensures the embedding payload. l_G^2 represents the adversarial loss of generator against all steganalyzers in the discriminator. The weights of the two components are denoted by α and β, respectively. We set $\alpha = 10^{-7}$ and $\beta = -1$ in our experiments.

The embedding loss l_G^1 is as follows:

$$l_G^1 = (-\sum_{\forall(i,j)} \sum_{\forall m} p_{i,j}^m \cdot log_2(p_{i,j}^m) - H \times W \times q)^2. \tag{8}$$

where m denotes the possible value of the embedding modifications in the specific color channel image (i.e., C_r, C_g or C_b), $m \in \{-1, 0, 1\}$. The embedding rate is denoted as q , with units in bpc (bits per channel pixel).

The adversarial loss l_G^2 is as follows:

$$l_G^2 = \frac{1}{H \times W} \cdot \sum_{\forall(i,j)} r_{i,j} \cdot ((log(p_{i,j}^{+1}) \cdot I(m_{i,j} = +1)) + (log(p_{i,j}^{-1}) \cdot I(m_{i,j} = -1))), \tag{9}$$

where $I(\cdot)$ is an indicator function. $r_{i,j}$ is the weight of the color component at position (i,j), and its formula is defined as follows:

$$r_{i,j} = \epsilon \cdot t_{i,j} \cdot m_{i,j} \cdot g_{i,j}, \tag{10}$$

where $\epsilon = 10^7$. $t_{i,j}$ is the absolute value of the image residual. It is obtained by applying the 8-neighborhood Laplacian operator to the specific color channel image. Here, the larger value of $t_{i,j}$ indicates that the texture is more complex at the pixel location. The gradient $g_{i,j}$ is defined as the partial derivative of the loss function L with respect to the modification $m_{i,j}$, and its formula is defined as follows:

$$g_{i,j} = \partial L / \partial m_{i,j}, \tag{11}$$

when the signs of $m_{i,j}$ and $g_{i,j}$ are consistent, the magnitude of $r_{i,j}$ acts as an incentive, encouraging the generator to modify the pixels in the specific color channel image so that the modification direction aligns with the gradient $g_{i,j}$, thereby disrupting the discriminator's ability to distinguish between the cover image and the stego image. The loss function L is defined as follows:

$$L = \min_{j \in \{1,\ldots,k\}} \{l_{D_j}(X, Y)\} + \omega_{min} \cdot l_{MAD}, \tag{12}$$

note that L consists two terms: the first term is the discrimination loss of the best-performing steganalyzer within the discriminator at each iteration, and the second term is the mean absolute deviation loss, wieghted by ω_{min}, which is defined by the following formula:

$$\omega_{min} = e^{(1 - \min_{j \in \{1,\ldots,k\}} \{l_{D_j}(X,Y)\})} / \sum_{j'=1}^{k} e^{(1 - l_{D_{j'}}(X,Y))}. \tag{13}$$

3 Experiments

In our experiments, we randomly selected 60,000 color images from the ALASKA II dataset [3]. Each image is in the spatial domain with a size of $256 \times 256 \times 3$ pixels. From these, 40,000 images were allocated for training the GAN model, while the remaining 20,000 images were reserved for testing. For performance evaluation, the test set was further divided into three disjoint subsets: 14,000 images for training the steganalysis models, 1,000 images for validation, and 5,000 images for final testing.

For comparison, we selected three traditional methods—CMD-C [22], ACMP [16], and GINA [23]—along with the GAN-based CIS-GAN [15]. Since the first three traditional methods rely on defining the initial costs based on grayscale steganography, we also included two classic grayscale techniques, S-UNIWARD [10] and HILL [13]. This resulted in six hybrid approaches for color steganography: CMD-C-SUNIWARD, CMD-C-HILL, ACMP-SUNIWARD, ACMP-HILL, GINA-SUNIWARD, and GINA-HILL. For security evaluation, six steganalytic methods were employed: one traditional method, SCRM [8], and five deep learning-based methods—CovNet [5], SR-Net [2], WISERNet [28], UCNet [25], and WeiNet [24].

The proposed method was implemented using the PyTorch. The experiments were conducted on a system equipped with an Intel(R) Core(TM) i7-6900K CPU operating at 3.20 GHz and four NVIDIA TITAN X GPUs. The batch sizes for the three training stages were set to 25, 25, and 20, respectively. The Adam optimizer was used for training, with a total of 72 epochs scheduled. Both the generator and discriminator were initialized with a learning rate of 0.0001, which was halved after every 20 epochs. [1]

3.1 Security Comparison with Related Methods

In this section, we present a comparative analysis of the proposed color steganographic method against seven alternative color steganographic techniques, evaluated using six distinct steganalysis methods. The results, shown in Table 1, demonstrate that our method consistently outperforms the alternatives on average, achieving state-of-the-art results in terms of steganographic security.

When compared to the second-best method - GINA-SUNIWARD, our method improves average security performance by 3.19% (from 27.75% to 30.94%) at an embedding rate of 0.4 bpc, and by 2.90% (from 38.54% to 41.44%) at an embedding rate of 0.2 bpc. Specifically, under analysis with SRNet at both the 0.4 bpc and 0.2 bpc embedding rates, our method shows significant security improvements of 10.27% (from 29.95% to 40.22%) and 5.36% (from 40.80% to 46.16%), respectively. Additionally, when compared to the GAN-based method CIS-GAN, our method significantly improves average performance by 18.79% (from 12.15% to 30.94%) at 0.4 bpc and by 22.85% (from 18.59% to 41.44%) at 0.2 bpc.

3.2 Comparison of Steganography Modification and Quality

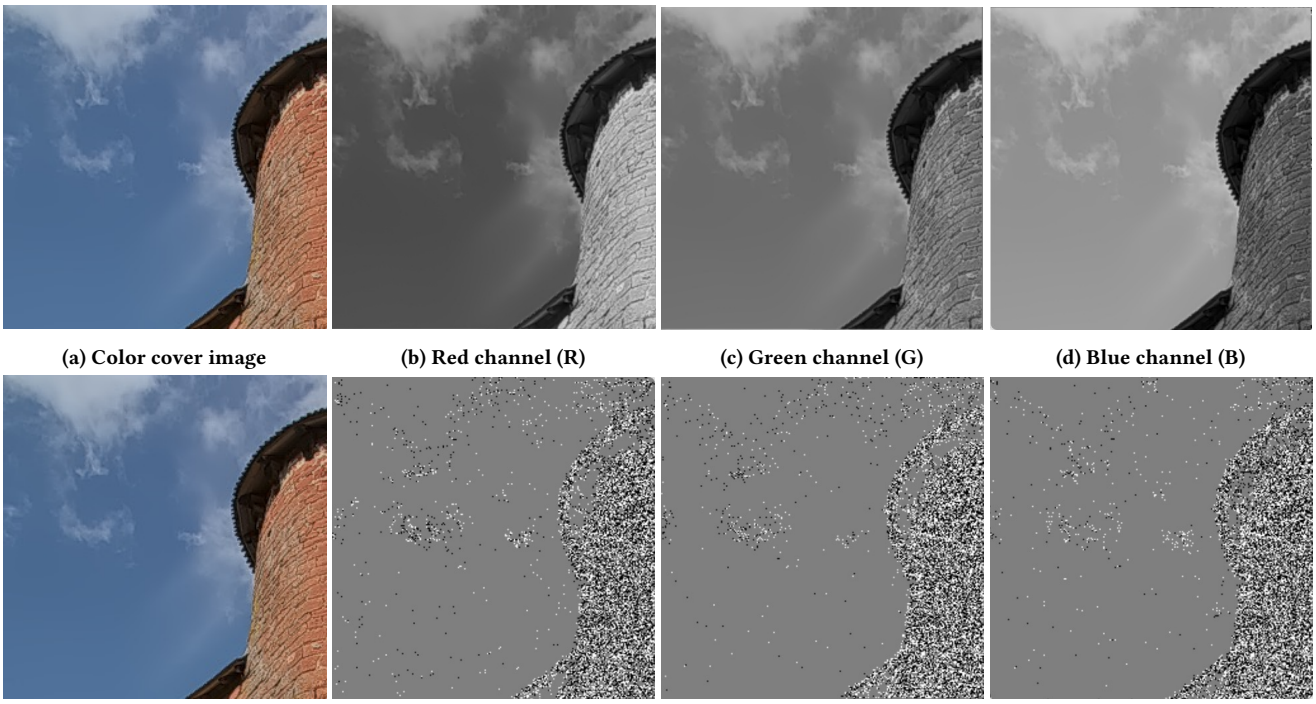
In this section, we calculated the modification rate and Peak Signal-to-Noise Ratio (PSNR) of the stego images using 20,000 test images at an embedding rate of 0.4bpc. The comparative results are presented in Table 2. These results demonstrate that our method maintains smaller and more balanced modification rates (relatively concentrated at 9.36%) across the three color channels, leading to stego images with higher PSNR. In comparison to the second-best method, GINA-SUNIWARD, the average modification rate decreased by 1.12% (from 10.48% to 9.36%), while the PSNR increased by 0.49 dB (from 57.93 dB to 58.42 dB). When compared to CIS-GAN, the average modification rate dropped by 2.19% (from 11.55% to 9.36%), and the PSNR rose by 0.91 dB (from 57.51 dB to 58.42 dB). Although our average modification rate is slightly higher than that of the ACMP-SUNIWARD, our security performance significantly surpasses it (refer to Table 1 for details).

Furthermore, we present the stego image and the modification maps for each channel after embedding information into an example image (Fig. 4-(a)) using our method. Fig. 4-(e) shows the stego image with a 0.4 bpc embedding payload. Distinguishing between

[1] https://github.com/Sanakkk3/Color-Image-Steganography-Using-Generative-Adversarial-Networks-with-a-Phased-Training-Strategy

Table 1: Detection error rate (%) evaluated on six steganalyzers on ALASKA. In the following tables, values marked with an asterisk (∗) and underlined indicate the best and second-best results for each case, respectively.

Embedding rate	Steganographic method	SCRM[8]	CovNet[5]	SRNet[2]	WISERNet[28]	UCNet[25]	WeiNet[24]	Average
0.4bpc	CMD-C-SUNIWARD[22]	26.32	33.55	24.04	23.85	19.95	18.80	24.42
	CMD-C-HILL[22]	28.05	29.44	23.80	25.04	17.15	16.55	23.34
	ACMP-SUNIWARD[16]	24.58	29.45	16.80	22.50	14.40	12.45	20.03
	ACMP-HILL[16]	26.39	21.51	13.17	20.52	10.48	9.65	16.95
	GINA-SUNIWARD[23]	29.26	36.10	29.95	26.37	**23.90***	20.90	27.75
	GINA-HILL[23]	30.37	32.38	23.20	28.70	18.50	17.15	25.05
	CIS-GAN[15]	20.93	21.41	5.97	9.47	7.24	7.90	12.15
	Proposed	**30.77***	**37.58***	**40.22***	**32.88***	23.60	**21.09***	**30.94***
0.2bpc	CMD-C-SUNIWARD[22]	36.20	41.70	33.40	36.32	27.63	25.20	33.41
	CMD-C-HILL[22]	37.58	37.15	31.60	39.55	21.95	21.30	31.52
	ACMP-SUNIWARD[16]	33.69	37.45	31.55	36.70	23.50	24.00	31.15
	ACMP-HILL[16]	36.22	35.20	25.60	38.25	20.45	18.80	29.09
	GINA-SUNIWARD[23]	38.20	**47.12***	40.80	43.10	32.70	29.30	38.54
	GINA-HILL[23]	39.37	38.15	33.70	40.50	26.54	24.25	33.75
	CIS-GAN[15]	29.68	30.96	11.59	11.84	12.09	15.35	18.59
	Proposed	**41.13***	46.21	**46.16***	**47.31***	**36.31***	**31.54***	**41.44***



(a) Color cover image **(b) Red channel (R)** **(c) Green channel (G)** **(d) Blue channel (B)**

(e) Color stego image (58.41dB) **(f) Modification map in R (9.58%)** **(g) Modification map in G (9.56%)** **(h) Modification map in B (9.31%)**

Figure 4: Visualization of an image example and the modification map for the three channels using the proposed method. The embedding rate is set to 0.4 bpc.

the cover and stego images is challenging for the human eye, as the PSNR of the stego image generated by our method reaches as high as 58.41 dB. Fig. 4-(b) to (d) are grayscale displays of the three color channels, and Fig. 4-(f) to (h) are the corresponding modification maps. In these maps, gray points indicate no embedding modification, white points signify a +1 modification, and and black points represent a −1 modification. This clearly demonstrates that our method effectively learns asymmetric embedding costs for

each color channel, focusing on challenging image textures, with minimal differences in modification positions across the channels. The modification rates differences between the channels are also minimal, with an average modification rate of 9.48%.

3.3 Comparison of Different Channel Embedding Orders

As discussed in Section 2-2.1, our proposed strategy embeds information into the three color channels of a color image in stages. In this section, we analyze the impact of different embedding orders on steganographic security. To this end, we evaluate all possible embedding sequences of the three color channels. The results are presented in Table 3. Experimental findings indicate that the embedding order significantly affects the model's security performance. When the proposed model follows the embedding order $r \rightarrow b \rightarrow g$, it achieves the highest average security performance. Compared to the second-best embedding order, $b \rightarrow g \rightarrow r$, the average security performance improves by 1.31% (from 31.70% to 33.01%). The embedding order $g \rightarrow b \rightarrow r$ exhibits the lowest security performance, with an average detection error rate of 28.51%.

3.4 Comparison of Different Steganalyzer Combinations in Discriminators

Based on our comparative experiments, we found that different combinations of steganalyzers used in discriminators for different color channels affect the final steganographic security performance. Specifically, using two steganalyzers for the red (R) and blue (B) channels, and three steganalyzers for the green (G) channel, gives better security performance. In this section, we examine various combinations of four commonly used steganalyzers: CovNet, WISERNet, UCNet, and a version of UCNet, called UCNet-v1. UCNet-v1 is a version of UCNet in which we reduced the number of convolutional layers based on an ablation study from the UCNet paper. As a result, UCNet-v1 is less effective at extracting steganalytic features and performs worse than UCNet.

Some comparative results are shown in Table 4. From this table, we observe that incorporating the stronger UCNet into the discriminator combination leads to relatively poorer performance. This is likely because the strong feedback from UCNet overwhelms the training process, causing instability. Based on our experiments, incorporating UCNet-v1 into D_g increases the average security performance by 2.67% (from 30.34% to 33.01%) compared to UCNet. Additionally, incorporating CovNet into D_r and WISERNet into D_b improves the average security performance by 2.12% (from 30.89% to 33.01%) and 1.97% (from 31.04% to 33.01%), respectively, achieving the highest security levels. Therefore, in the proposed model, CovNet and WISERNet are used in D_r and D_b, while CovNet, WISERNet, and UCNet-v1 are used in D_g.

3.5 Impact of Mean Absolute Deviation Loss

As described in Section 2- 2.5, we designed a mean absolute deviation loss function, l_{MAD}, combined with weights ω_{max} and ω_{min} to implement our proposed update strategy (see Formulas (5) and (12)). In this section, we will examine the impact of l_{MAD} on the final performance. To this end, the following three variants are included for comparative study.

- **Variant 1**: Set both ω_{max} and ω_{min} to 0, meaning that l_{MAD} is not used at all. In this case, the generator's parameters are optimized based solely on feedback from the best-performing steganalyzer, while the weakest steganalyzer is also optimized (i.e., the strategy shown in Fig.3-(a)).
- **Variant 2**: Set ω_{max} to 0 in formula (5). In this case, feedback from all steganalyzers is dynamically weighted to optimize the generator, while optimizing the weakest steganalyzer.
- **Variant 3**: Set ω_{min} to 0 in formula (12). In this case, the generator is optimized using feedback from the best-performing steganalyzer, while all steganalyzers are optimized simultaneously.

The comparative results shown in Table 5 indicate that the security performance of the proposed model is significantly influenced by the l_{MAD} loss function. Compared to Variant 1, Variant 2, and Variant 3, the proposed method achieves improvements of 2.77%, 2.00%, and 1.57% in average security performance, respectively. These results confirm that the proposed update strategy facilitates the progressive improvement of both the generator and discriminator's performance and contributes to stego images with enhanced resistance to steganalysis detection.

3.6 Comparison of Phased and Simultaneous Embedding Strategies

As described in Section 2-2.1, we adopt a phased training and embedding strategy for the different color channels. In this section, we compare the phased strategy with an alternative method, referred to as Strategy 1. In Strategy 1, the generator simultaneously trains all three color channels and performs the embedding for all channels at the same time during the embedding stage.

The experimental results are presented in Table 6. Notably, our proposed model consistently outperforms Strategy 1 in terms of steganography security across all cases. Specifically, the proposed method shows an improvement of 6.75% in average security performance. These findings confirm that the phased embedding strategy effectively leverages the correlation between color channels, leading to stego images with enhanced resistance to steganalysis detection.

3.7 Experiments on Another Dataset

To further validate the effectiveness of our method, we conducted comparative experiments on another dataset, BOSSBase [1]. We processed 40,000 color images of size 256×256 from the BOSSBase dataset and randomly split them into two sets: 32,000 images for training and 8,000 images for testing. The test set was then randomly divided into three disjoint subsets: 4,800 images for training, 800 images for validation, and 2,400 images for testing.

The comparative results presented in Table 7 demonstrate that the proposed method still achieves the best security performance to date. Compared to the second-best performing method (i.e., GINA-HILL; note that GINA-SUNIWARD is the second-best for the ALASKA dataset), our average performance improved by 5.01% (from 18.71% to 23.72%) at 0.4 bpc and 5.64% (from 30.12% to 35.76%) at 0.2 bpc. Specifically, under analysis with WeiNet at both the 0.4 bpc and 0.2 bpc embedding rates, our method shows significant security improvements of 6.31% (from 11.75% to 18.06%) and 15.02% (from 18.31% to 33.33%), respectively. Additionally, When compared

Table 2: The modification rate of various steganographic methods and the corresponding PSNR of stego images. In this experiment, the embedding rate is set to 0.4 bpc.

Steganographic method	Channel **R**	Channel **G**	Channel **B**	Average	PSNR(dB)
CMD-C-SUNIWARD[22]	9.87%	11.50%	12.44%	11.27%	57.62
CMD-C-HILL[22]	9.88%	11.29%	12.16%	11.11%	57.67
ACMP-SUNIWARD[16]	**8.88%***	7.80%	**8.70%***	**8.46%***	**58.86***
ACMP-HILL[16]	10.66%	8.32%	10.40%	9.79%	58.22
GINA-SUNIWARD[23]	11.56%	8.66%	11.21%	10.48%	57.93
GINA-HILL[23]	12.48%	7.74%	11.90%	10.71%	57.83
CIS-GAN[15]	14.61%	**7.21%***	12.82%	11.55%	57.51
Proposed	9.39%	9.48%	9.21%	9.36%	58.42

Table 3: Detection error rate (%) of the proposed method with different embedding orders. In this experiment, the embedding rate is set to 0.4 bpc.

Embedding order	SCRM[8]	CovNet[5]	SRNet[2]	WISERNet[28]	UCNet[25]	Average
$g \to r \to b$	28.59	34.35	35.15	28.10	17.90	28.82
$g \to b \to r$	28.84	33.80	33.30	27.80	18.80	28.51
$b \to r \to g$	30.28	34.50	36.05	30.70	19.40	30.19
$b \to g \to r$	32.19	36.15	37.35	**33.70***	19.10	31.70
$r \to g \to b$	**33.10***	**38.25***	35.30	28.15	19.95	30.95
Proposed ($r \to b \to g$)	30.77	37.58	**40.22***	32.88	**23.60***	**33.01***

Table 4: Detection error rate (%) of the proposed method with different combinations of steganalyzers in discriminators. In this experiment, the embedding rate is set to 0.4 bpc.

Configuration (D_r and D_b)	Configuration (D_g)	SCRM[8]	CovNet[5]	SRNet[2]	WISERNet[28]	UCNet[25]	Average
WISERNet and UCNet	CovNet, WISERNet, and UCNet	30.49	35.10	35.25	28.85	20.05	29.95
CovNet and UCNet		30.68	32.65	36.30	27.45	20.60	29.54
CovNet and WISERNet		31.13	34.65	36.75	28.85	20.30	30.34
WISERNet and UCNet-v1	**CovNet, WISERNet, and UCNet-v1**	31.32	34.75	37.90	29.70	20.80	30.89
CovNet and UCNet-v1		**31.87***	36.10	35.64	29.85	21.75	31.04
CovNet and WISERNet		30.77	**37.58***	**40.22***	**32.88***	**23.60***	**33.01***

Table 5: Detection error rate (%) of the proposed method with different introduced mean absolute deviation loss strategies. In this experiment, the embedding rate is set to 0.4 bpc.

Steganalyzer	Variant 1	Variant 2	Variant 3	**Proposed**
SCRM[8]	**31.31***	30.97	30.51	30.77
CovNet[5]	34.70	35.88	35.85	**37.58***
SRNet[2]	36.35	38.60	38.48	**40.22***
WISERNet[28]	30.80	29.62	30.57	**32.88***
UCNet[25]	18.05	19.97	21.78	**23.60***
Average	30.24	31.01	31.44	**33.01***

Table 6: Detection error rate (%) of the proposed method with different embedding strategies. In this experiment, the embedding rate is set to 0.4 bpc.

Steganalyzer	Strategy 1	**Proposed**
SCRM[8]	30.31	**30.77***
CovNet[5]	28.75	**37.58***
SRNet[2]	32.10	**40.22***
WISERNet[28]	21.90	**32.88***
UCNet[25]	18.25	**23.60***
Average	26.26	**33.01***

to the existing GAN-based method, CIS-GAN, our method shows an improvement of 15.13% (from 8.59% to 23.72%) at 0.4 bpc and 24.52% (from 11.24% to 35.76%) at 0.2 bpc.

4 Conclusion

In this paper, we propose a Color Image Steganography method using GANs that learns the asymmetric embedding costs across different color channels from scratch. We investigate how the order

Table 7: Detection error rate (%) evaluated on six steganalyzers on another image dataset-BOSSBase.

Embedding rate	Steganographic method	SCRM[8]	CovNet[5]	SRNet[2]	WISERNet[28]	UCNet[25]	WeiNet[24]	Average
0.4bpc	CMD-C-SUNIWARD[22]	17.27	15.40	15.19	12.54	11.06	8.56	13.34
	CMD-C-HILL[22]	18.42	16.98	10.81	14.00	9.21	8.13	12.93
	ACMP-SUNIWARD[16]	15.60	14.25	8.46	10.58	5.63	3.87	9.73
	ACMP-HILL[16]	16.29	11.75	9.21	9.67	6.96	3.19	9.51
	GINA-SUNIWARD[23]	21.63	16.35	17.79	16.39	13.15	11.44	16.13
	GINA-HILL[23]	**25.21***	20.65	20.65	19.54	14.46	11.75	18.71
	CIS-GAN[15]	20.46	12.92	3.00	2.40	5.63	7.15	8.59
	Proposed	24.63	**22.37***	**26.95***	**29.01***	**21.29***	**18.06***	**23.72***
0.2bpc	CMD-C-SUNIWARD[22]	28.17	22.81	22.12	27.44	17.63	16.25	22.40
	CMD-C-HILL[22]	32.92	25.25	16.75	32.87	16.63	13.17	22.93
	ACMP-SUNIWARD[16]	27.69	22.75	18.50	32.85	14.56	13.44	21.63
	ACMP-HILL[16]	28.96	23.00	16.63	31.25	13.50	10.50	20.64
	GINA-SUNIWARD[23]	34.17	24.31	24.38	41.25	20.56	19.69	27.39
	GINA-HILL[23]	34.60	31.15	26.31	47.75	22.62	18.31	30.12
	CIS-GAN[15]	26.52	15.15	4.71	3.25	7.56	10.25	11.24
	Proposed	**34.76***	**33.79***	**33.27***	**48.13***	**31.27***	**33.33***	**35.76***

of embedding across different color channels affects steganographic security and introduce a phased training and embedding process that better captures the inter-channel relationships, thereby enhancing the overall security. Additionally, we design an adaptive model update strategy and introduce a novel loss function, the Mean Absolute Deviation (MAD), which dynamically balances the GAN training process. This improvement enhances the generator's steganographic capabilities. Extensive experiments on two color image datasets demonstrate that our method significantly outperforms existing approaches. Ablation studies further validate the rationale of the proposed model.

Despite these advancements, our method still face challenges in model optimization and adaptation to various application scenarios. First, the phased training approach introduced in this paper results in relatively low time efficiency, which will be a focus for future research aimed at improving training efficiency. Second, while the proposed method is designed for spatial domain color images, JPEG-format color images are more commonly used in practical applications. Therefore, future work will investigate the characteristics of JPEG color images and extend the proposed method to meet the requirements of the JPEG domain.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62472458 and No. U22A2030), Guangdong Provincial Key Laboratory of Information Security Technology (No. 2025A1515012823), and Hunan Provincial Funds for Distinguished Young Scholars (No. 2024JJ2025).

References

[1] Patrick Bas, Tomáš Filler, and Tomáš Pevnỳ. 2011. " Break our steganographic system": the ins and outs of organizing BOSS. In *International workshop on information hiding*. Springer, 59–70.

[2] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. 2018. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security* 14, 5 (2018), 1181–1193.

[3] Rémi Cogranne, Éva Giboulot, and Patrick Bas. 2020. ALASKA# 2: Challenging academic research on steganalysis with realistic images. In *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 1–5.

[4] Tomáš Denemark and Jessica Fridrich. 2015. Improving steganographic security by synchronizing the selection channel. In *Proceedings of the 3rd ACM workshop on information hiding and multimedia security*. 5–14.

[5] Xiaoqing Deng, Bolin Chen, Weiqi Luo, and Da Luo. 2019. Fast and effective global covariance pooling network for image steganalysis. In *Proceedings of the ACM workshop on information hiding and multimedia security*. 230–234.

[6] Tomáš Filler, Jan Judas, and Jessica Fridrich. 2011. Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Transactions on Information Forensics and Security* 6, 3 (2011), 920–935.

[7] Jessica Fridrich and Tomas Filler. 2007. Practical methods for minimizing embedding impact in steganography. In *Security, Steganography, and Watermarking of Multimedia Contents IX*, Vol. 6505. SPIE, 13–27.

[8] Miroslav Goljan, Jessica Fridrich, and Rémi Cogranne. 2014. Rich model for steganalysis of color images. In *2014 IEEE International workshop on information forensics and security (WIFS)*. IEEE, 185–190.

[9] Vojtěch Holub and Jessica Fridrich. 2012. Designing steganographic distortion using directional filters. In *2012 IEEE International workshop on information forensics and security (WIFS)*. IEEE, 234–239.

[10] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. 2014. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security* 2014 (2014), 1–13.

[11] Dongxia Huang, Weiqi Luo, Peijia Zheng, and Jiwu Huang. 2023. Automatic asymmetric embedding cost learning via generative adversarial networks. In *Proceedings of the 31st ACM International Conference on Multimedia*. 8316–8326.

[12] Ying Jin, Yunbo Wang, Mingsheng Long, Jianmin Wang, S Yu Philip, and Jiaguang Sun. 2020. A multi-player minimax game for generative adversarial networks. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1–6.

[13] Bin Li, Ming Wang, Jiwu Huang, and Xiaolong Li. 2014. A new cost function for spatial image steganography. In *2014 IEEE International conference on image processing (ICIP)*. IEEE, 4206–4210.

[14] Bin Li, Ming Wang, Xiaolong Li, Shunquan Tan, and Jiwu Huang. 2015. A strategy of clustering modification directions in spatial image steganography. *IEEE Transactions on Information Forensics and Security* 10, 9 (2015), 1905–1917.

[15] Xin Liao, Zhiqiang Tang, and Yun Chao. 2021. Steganographic Distortion Function Design Method for Spatial Color Image Based on GAN. *Journal of Software* 33, 9 (2021), 3470–3484.

[16] Xin Liao, Yingbo Yu, Bin Li, Zhongpeng Li, and Zheng Qin. 2019. A new payload partition strategy in color image steganography. *IEEE transactions on circuits and systems for video technology* 30, 3 (2019), 685–696.

[17] Weiqi Luo, Kangkang Wei, Qiushi Li, Miaoxin Ye, Shunquan Tan, Weixuan Tang, Jiwu Huang, et al. 2024. A Comprehensive Survey of Digital Image Steganography and Steganalysis. *APSIPA Transactions on Signal and Information Processing* 13, 1 (2024).

[18] Tomáš Pevnỳ, Tomáš Filler, and Patrick Bas. 2010. Using high-dimensional image models to perform highly undetectable steganography. In *Information Hiding:*

12th International Conference, IH 2010, Calgary, AB, Canada, June 28-30, 2010, Revised Selected Papers 12. Springer, 161–177.

[19] Xinghong Qin, Bin Li, Shunquan Tan, Weixuan Tang, and Jiwu Huang. 2022. Gradually enhanced adversarial perturbations on color pixel vectors for image steganography. *IEEE Transactions on Circuits and Systems for Video Technology* 32, 8 (2022), 5110–5123.

[20] Xinghong Qin, Bin Li, Shunquan Tan, and Jishen Zeng. 2019. A novel steganography for spatial color images based on pixel vector cost. *IEEE Access* 7 (2019), 8834–8846.

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 234–241.

[22] Weixuan Tang, Bin Li, Weiqi Luo, and Jiwu Huang. 2015. Clustering steganographic modification directions for color components. *IEEE Signal Processing Letters* 23, 2 (2015), 197–201.

[23] Yaofei Wang, Weiming Zhang, Weixiang Li, Xinzhi Yu, and Nenghai Yu. 2019. Non-additive cost functions for color image steganography based on inter-channel correlations and differences. *IEEE Transactions on Information Forensics and Security* 15 (2019), 2081–2095.

[24] Kangkang Wei, Weiqi Luo, and Jiwu Huang. 2024. Color Image Steganalysis Based on Pixel Difference Convolution and Enhanced Transformer With Selective Pooling. *IEEE Transactions on Information Forensics and Security* (2024).

[25] Kangkang Wei, Weiqi Luo, Shunquan Tan, and Jiwu Huang. 2022. Universal deep network for steganalysis of color image based on channel representation. *IEEE Transactions on Information Forensics and Security* 17 (2022), 3022–3036.

[26] Wei xiang Li, Weiming Zhang, Li Li, Hang Zhou, and Nenghai Yu. 2020. Designing Near-Optimal Steganographic Codes in Practice Based on Polar Codes. *IEEE Transactions on Communications* 68 (2020), 3948–3962.

[27] Guanshuo Xu, Han-Zhou Wu, and Yun-Qing Shi. 2016. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters* 23, 5 (2016), 708–712.

[28] Jishen Zeng, Shunquan Tan, Guangqing Liu, Bin Li, and Jiwu Huang. 2019. WISERNet: Wider separate-then-reunion network for steganalysis of color images. *IEEE Transactions on Information Forensics and Security* 14, 10 (2019), 2735–2748.