

Spatial Image Steganography Based on Generative Adversarial Network

Jianhua Yang, Kai Liu, *Student Member, IEEE*, Xiangui Kang*, *Senior Member, IEEE*, Edward K.Wong, *Senior Member, IEEE*, Yun-Qing Shi, *Life Fellow, IEEE*

Abstract—With the recent development of deep learning on steganalysis, embedding secret information into digital images faces great challenges. In this paper, a secure steganography algorithm by using adversarial training is proposed. The architecture contain three component modules: a generator, an embedding simulator and a discriminator. A generator based on U-NET to translate a cover image into an embedding change probability is proposed. To fit the optimal embedding simulator and propagate the gradient, a function called Tanh-simulator is proposed. As for the discriminator, the selection-channel awareness (SCA) is incorporated to resist the SCA based steganalytic methods. Experimental results have shown that the proposed framework can increase the security performance dramatically over the recently reported method ASDL-GAN, while the training time is only 30% of that used by ASDL-GAN. Furthermore, it also performs better than the hand-crafted steganographic algorithm S-UNIWARD.

Index Terms—Steganography, Steganalysis, Generative adversarial network (GAN).

I. INTRODUCTION

Image steganography is a kind of technology to embed secret information into a cover image without drawing suspicion. With the development of steganalysis methods, it becomes a great challenge to design a secure steganographic scheme. Because the efficient coding schemes [1] can embed messages close to the payload-distortion bound, the main task of image steganography is to minimize a well designed additive distortion function. In an adaptive steganography scheme, every pixel is assigned a cost to quantify the effect of making modification and the distortion is evaluated by summing up costs. Secret information is generally embedded in noisy regions or regions with texture, while smooth regions are avoided for data embedding, as done by HUGO [2], WOW [3], HILL [4], S-UNIWARD [5] and MiPOD [6].

In recent years, convolutional neural networks (CNN) have become a dominant machine learning approach in image classification tasks with the improvements in computer hardware and network architecture [7, 8]. Current researches have

This work was supported by NSFC (Grant Nos. U1536204, 61772571, 61702429), and the special funding for basic scientific research of Sun Yat-sen University (6177060230). (Corresponding author: Xiangui Kang.)

J. Yang, K. Liu, X. Kang are with Guangdong Key Lab of Information Security, School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, China 510006, (e-mail: isskxg@mail.sysu.edu.cn).

E. K. Wong is with Department of Computer Science and Engineering, New York University, Tandon School of Engineering, Brooklyn, NY 11201, (e-mail:ewong@nyu.edu).

Y. Shi is with Department of ECE, New Jersey Institute of Technology, Newark, NJ, USA 07102, (e-mail:shi@njit.edu).

indicated that CNN also obtained considerable achievements in the field of steganalysis. Tan and Li [9] used the stacked convolutional auto-encoder for steganalysis. Qian *et al* [10, 11] proposed a CNN structure equipped with Gaussian non-linear activation, and they showed that feature representations can be transferred from high embedding payload to low embedding payload. Xu *et al* [12, 13] proposed a CNN structure (referred to as XuNet in this paper) that is able to achieve comparable performance to conventional spatial rich model (SRM) [14]. The Tanh and ReLU have been used in shallow and deep layers respectively. Batch-normalization [15] was equipped to prevent the network from falling into local minima. Yang *et al* [16] incorporated selection-channel awareness (SCA) into the CNN architecture. Ye *et al* [17] proposed a structure that incorporates high-pass filters from SRM, the SCA also be incorporated in CNN architecture. In [18], Yang *et al* proposed a deep learning architecture by improving the pre-processing layer and the feature reuse for JPEG steganalysis, experimental results shows that it can obtain state-of-the art performance for JPEG steganalysis. Although CNN has been successfully used for steganalysis, it is still in initial stage with regarding to applying it for steganography.

So far, the generative adversarial network (GAN) [19] has been widely used for image generation [20, 21]. In [22], Tang *et al* proposed an automatic steganographic distortion learning framework with GAN (named as ASDL-GAN shortly). The probability of data embedding is learned via the adversarial training between the generator and the discriminator. The generator contains 25 groups, with every group starts with a convolutional layer, followed by batch normalization and a ReLU layer. The architecture of XuNet was employed as the discriminator. In order to fit the optimal embedding simulator [23] as well as propagate the gradient in back propagation, they proposed a ternary embedding simulator (TES) activation function. The reported experimental results showed that ASDL-GAN can learn steganographic distortions, but the performance is still inferior to the conventional steganographic scheme S-UNIWARD.

In this work, we propose a new GAN-based steganographic framework. Compared with the previous method ASDL-GAN [22], the main contributions of this paper are as follows.

- (1) An activation function called Tanh-simulator is proposed to solve the problem that optimal embedding simulator cannot propagate gradient. The TES sub-network of ASDL-GAN needs a long time to pre-train with even 10^6 iterations, while the Tanh-simulator can be used directly with high fitting accuracy.

- (2) A more compact generator based on U-NET [24] has been proposed. This subnet can improves security performance and decreases training time dramatically.
- (3) Considering adversarial training, we enhance the discriminator by incorporating SCA into the discriminator to improve the performance of resisting SCA based steganalytic schemes.

The rest of the paper is organized as follows. The proposed architecture is described in Section II. Experimental results and analysis is shown in Section III. The practical application of the proposed architecture is shown in Section IV. The conclusion and future works are presented in Section V.

II. THE PROPOSED ARCHITECTURE

In this section, firstly, we present the overall architecture of the proposed method based on generative adversarial network (referred as UT-SCA-GAN), which incorporates the U-net based generator, the proposed Tanh-simulator function and the SCA based discriminator. Secondly, the definition of the loss functions is introduced. Then, the details of the generator and the proposed Tanh-simulator function are described. Finally, we present the design consideration of the discriminator.

A. Overall Architecture

The proposed overall architecture is shown in Fig. 1. The training steps are described as follows:

- (1) Translate a cover image into an embedding change probability map by using the generator.
- (2) Given an embedding change probability map and a randomly generated matrix with uniform distribution of [0,1], compute the modification map by using the proposed Tanh-simulator.
- (3) Generate the stego image by adding the cover image and its corresponding modification map.
- (4) Feed cover/stego pairs and the corresponding embedding change probability map into the discriminator.
- (5) Alternately update the parameters of generator and discriminator.

B. The Loss Functions

The loss function of the discriminator is defined as follows:

$$l_D = - \sum_{i=1}^2 y'_i \log(y_i) \quad (1)$$

where y_i is the Softmax output of the discriminator, while y'_i is the corresponding truth label of cover/stego.

The loss function of the generator is defined as follows [22]:

$$l_G = -\alpha \times l_D + \beta \times (C - H \times W \times Q)^2 \quad (2)$$

where H and W are the height and width of the cover image, Q denotes the embedding payload, and C is the capacity that guarantees the payloads:

$$C = \sum_{i=1}^H \sum_{j=1}^W -p_{i,j}^{+1} \log_2 p_{i,j}^{+1} - p_{i,j}^{-1} \log_2 p_{i,j}^{-1} - p_{i,j}^0 \log_2 p_{i,j}^0 \quad (3)$$

$$p_{i,j}^{-1} = p_{i,j}^{+1} = p_{i,j}/2 \quad (4)$$

$$p_{i,j}^0 + p_{i,j}^{-1} + p_{i,j}^{+1} = 1 \quad (5)$$

where $p_{i,j}$ denotes the output embedding probability of the generator corresponding to the pixel $x_{i,j}$, $p_{i,j}^{+1}$ and $p_{i,j}^{-1}$ denote the modify probability of adding or subtracting 1, while $p_{i,j}^0$ denote the probability of the corresponding pixel $x_{i,j}$ will not be modified.

C. Generator Design

Motivated by an elegant architecture “U-Net” [24], which was used for biomedical image segmentation, we design an efficient generator for secure steganography based on U-Net. A typical architecture of U-Net is shown in Fig. 2. The detailed configuration of the proposed generator is shown in Table I. Note that in the contracting path, each group shown in the table corresponds to the sequence of convolution, batch normalization and Leaky-ReLu. A group of the expanding path corresponds to the sequence of deconvolution, batch normalization and ReLu. The last layer ensures that the embedding probability ranges from 0 to 0.5 by considering large embedding probability may caused the embedding process be easily detected [25]. The Leaky-ReLu activation function is defined as follows:

$$f(x) = \begin{cases} x & x > 0 \\ \alpha x & x \leq 0 \end{cases} \quad (6)$$

To prevent the “dying ReLU” problem, we set $\alpha = 0.2$. The main characteristics of the generator are described as follows:

- (1) It is composed of the contracting path and the expanding path. The former follows the typical architecture of a convolutional neural network while the latter mainly consists of deconvolution operations.
- (2) In order to achieve pixel-level learning and facilitate the back-propagation of gradients, there are concatenation connections between every pair of mirrored layers with the same size, such as layers 1 and 15, layers 2 and 14, etc.
- (3) The middle layers capture the global information of the image, while both sides of the generator provide local information.

Different from the 25-layer generator in ASDL-GAN [22], here the pre-processing layer is not used. In addition, the generator converges quickly and trains faster due to skipping connections and low memory consumption.

D. Tanh-simulator Function

In the previous adaptive steganography methods [2–6], stego image is generated by adding the cover image and the corresponding modification map. The modification map is computed by using an optimal embedding simulator [23], which is a staircase function:

$$m_{i,j} = \begin{cases} -1 & \text{if } n_{i,j} < p_{i,j}/2 \\ 1 & \text{if } n_{i,j} > 1 - p_{i,j}/2 \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

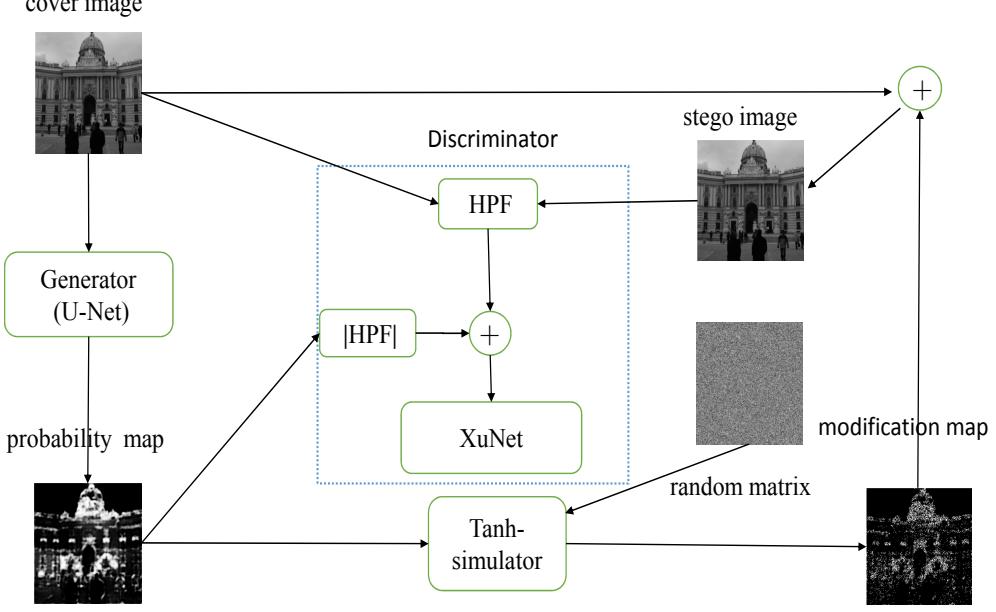


Fig. 1: Steganographic architecture of the proposed UT-SCA-GAN.

TABLE I: Configuration details of the proposed generator.

Layers	Output size	Kernels size	Process
Input	$1 \times (512 \times 512)$	/	Convolution-BN-Leaky ReLU
Layer 1	$16 \times (256 \times 256)$	$16 \times (3 \times 3 \times 1)$	Convolution-BN-Leaky ReLU
Layer 2	$32 \times (128 \times 128)$	$32 \times (3 \times 3 \times 16)$	Convolution-BN-Leaky ReLU
Layer 3	$64 \times (64 \times 64)$	$64 \times (3 \times 3 \times 32)$	Convolution-BN-Leaky ReLU
Layer 4	$128 \times (32 \times 32)$	$128 \times (3 \times 3 \times 64)$	Convolution-BN-Leaky ReLU
Layer 5	$128 \times (16 \times 16)$	$128 \times (3 \times 3 \times 128)$	Convolution-BN-Leaky ReLU
Layer 6	$128 \times (8 \times 8)$	$128 \times (3 \times 3 \times 128)$	Convolution-BN-Leaky ReLU
Layer 7	$128 \times (4 \times 4)$	$128 \times (3 \times 3 \times 128)$	Convolution-BN-Leaky ReLU
Layer 8	$128 \times (2 \times 2)$	$128 \times (3 \times 3 \times 128)$	Convolution-BN-Leaky ReLU
Layer 9	$256 \times (4 \times 4)$	$128 \times (5 \times 5 \times 128)$	Deconvolution-BN-ReLU-Concatenate
Layer 10	$256 \times (8 \times 8)$	$128 \times (5 \times 5 \times 128)$	Deconvolution-BN-ReLU-Concatenate
Layer 11	$256 \times (16 \times 16)$	$128 \times (5 \times 5 \times 256)$	Deconvolution-BN-ReLU-Concatenate
Layer 12	$256 \times (32 \times 32)$	$128 \times (5 \times 5 \times 256)$	Deconvolution-BN-ReLU-Concatenate
Layer 13	$128 \times (64 \times 64)$	$64 \times (5 \times 5 \times 256)$	Deconvolution-BN-ReLU-Concatenate
Layer 14	$64 \times (128 \times 128)$	$32 \times (5 \times 5 \times 128)$	Deconvolution-BN-ReLU-Concatenate
Layer 15	$32 \times (256 \times 256)$	$16 \times (5 \times 5 \times 64)$	Deconvolution-BN-ReLU-Concatenate
Layer 16	$1 \times (512 \times 512)$	$1 \times (5 \times 5 \times 32)$	Deconvolution-BN-ReLU-Concatenate
Layer 17	$1 \times (512 \times 512)$	/	ReLU((Sigmoid-0.5))

where $p_{i,j}$ is the embedding change probability, $n_{i,j}$ is the random number generated from a uniform distribution between 0 and 1, $m_{i,j}$ is the embedding value.

Because the staircase function (7) cannot convey the gradient during back propagation, we use the Tanh function to fit the embedding simulator. We called the proposed activation function Tanh-simulator. It can be described as follows:

$$m'_{i,j} = -0.5 \times \tanh(\lambda(p_{i,j} - 2 \times n_{i,j})) + 0.5 \times \tanh(\lambda(p_{i,j} - 2 \times (1 - n_{i,j}))) \quad (8)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (9)$$

where λ is the scaling factor, which controls the slope at the junction of stairs. Fig. 3 and Fig. 4 illustrate the function

curves of the Tanh-simulator and the staircase function (7) in 2D and 3D respectively. It can be seen that with the increment of parameter λ , Tanh-simulator is becoming more similar to the staircase function (7). Note that we need some discrete points to convey the gradient, considering the conveyance of gradient and fitting accuracy, we set $\lambda = 1000$.

E. Discriminator Design

Discriminator is a steganalytic tool in this framework. Considering the adversarial training between steganography and steganalysis, we infer that the enhancement of the discriminator will certainly force the steganography to be more secure.

To resist the current SCA based steganalysis method, we incorporate SCA into the discriminator. The $|HPF|$ in Fig. 1

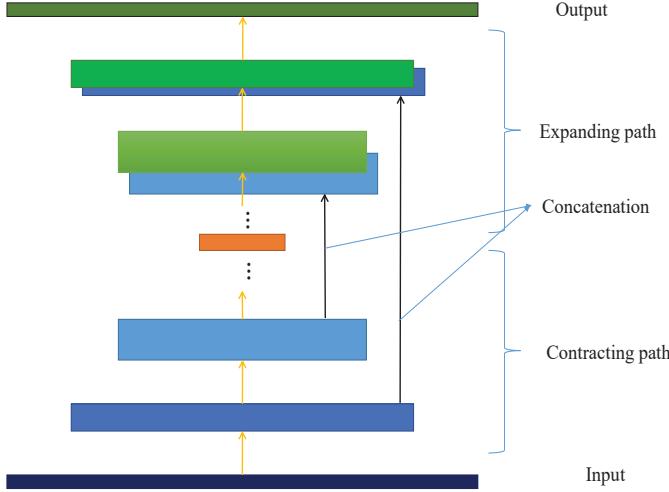


Fig. 2: Typical architecture of the U-Net.

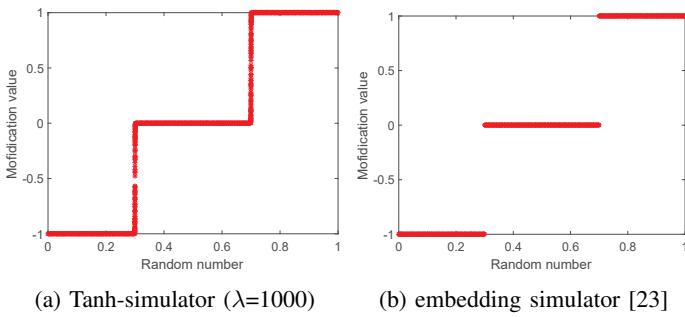


Fig. 3: Function curves of the Tanh-simulator and the embedding simulator in 2D space ($p_{i,j} = 0.6$): (a) Tanh-simulator ($\lambda=1000$), (b) embedding simulator [23].

denote the absolute value of the 30 high pass filters from SRM to consider the statistical measure [17]. Through adversarial training, the generator will adjust the parameters to resist the SCA based discriminator. In addition, the embedding change probability map bypasses the Tanh-simulator will accelerate the gradient propagation in adversarial training.

III. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Setting

All experiments are conducted on the SZUBase [22] and BOSSBase-1.01 [26] which contains images with size of 512×512 . The first dataset that contains 40,000 grayscale cover images is used to train the proposed architecture. The second dataset that contains 10,000 grayscale images is used to generate stego images. 5,000 pairs of images from BOSSBase are randomly selected to train the ensemble classifier [27], and the left 5,000 pairs are used as the test set. We use the Adam optimizer with the learning rate of 0.0001 to train the model over 160,000 iterations (32 epochs). During training, 8 cover/stego pairs are used as input in each iteration. The parameters α , β are set to 1, and 10^{-7} , respectively. All experiments are performed with TensorFlow on a GTX 1080 Ti GPU card.

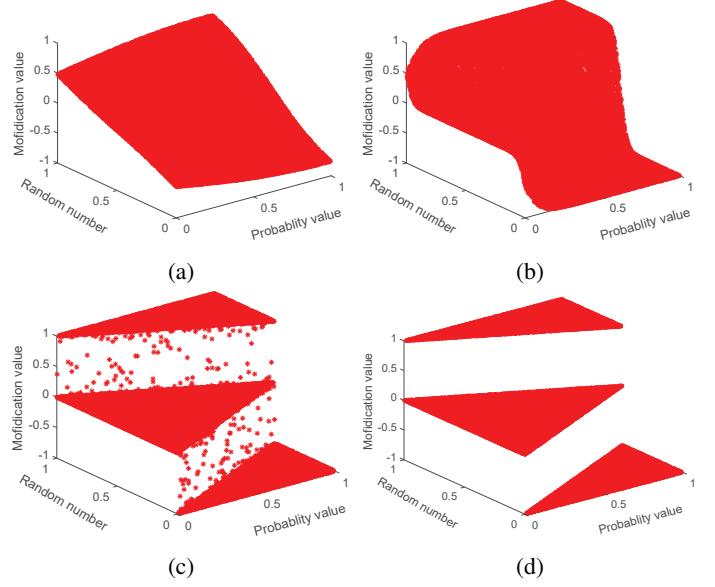


Fig. 4: Comparison between Tanh-simulator and embedding simulator in 3D space: (a) Tanh-simulator ($\lambda=1$), (b) Tanh-simulator ($\lambda=10$), (c) Tanh-simulator ($\lambda=1000$), (d) embedding simulator [23].

B. Experiments on resized dataset

In this part, we conduct experiments to investigate the influence of different parts of the proposed architecture. The original dataset of SZUBase and BOSSbase-1.01 are resized to 256×256 by “imresize()” function in Matlab.

Firstly, we conduct experiments on UT-GAN, which is the variant of UT-SCA-GAN without SCA. We vary the UT-GAN as follows:

- (1) Variant #1, replace the proposed generator with the generator of ASDL-GAN.
- (2) Variant #2, replace the Tanh-Simulator with the TES network of ASDL-GAN.

Experimental results tested on BOSSBase 256×256 are shown on Table II. All of the methods are embedded messages with payload 0.4 bpp. From Table II, it can be seen that the performance will decrease dramatically if we replace the generator or embedding simulator with the corresponding parts of the ASDL-GAN. Replacement of the generator caused the greatest performance reduction, it is because the proposed generator directly determines the adaptability and security of the steganographic scheme. The Tanh-simulator also achieves better performance than the TES network, the result might be caused by high fitting accuracy. In addition, the proposed UT-GAN also obtains better performance than ASDL-GAN and S-UNIWARD.

Next, we conduct experiments with the UT-SCA-GAN which incorporates SCA in the discriminator to verify the influence of SCA. The payload of UT-SCA-GAN and UT-GAN are set as 0.4 bpp. We test the performance by SRM and maxSRMd2. Experimental results are show on Table III.

As we expected, the incorporation of SCA to the discriminator will improve the security performance by about 2.0% to resist SCA based steganalysis method, e.g. maxSRMd2. Because

TABLE II: Error rates (%) of different steganographic schemes detected by SRM [14] on BOSSbase 256×256 .

Algorithm	UT-GAN	Variant #1	Variant #2	ASDL-GAN [22]	S-UNIWARD[5]
Error rates	26.61	20.29	23.06	20.68	22.26

the SCA is incorporated into the discriminator, the parameters of generator can be automatically adjusted according to the structure of discriminator via the adversarial training.

TABLE III: Error rates (%) of UT-GAN and UT-SCA-GAN on BOSSBase 256×256 .

Net-work	maxSRMd2	SRM
UT-GAN	20.27	26.61
UT-SCA-GAN	22.30	26.43

C. Experiments on full size dataset

In this part, we conduct experiments on size of 512×512 to compare with previous method. For 0.1 bpp, we finetune the architecture from the trained model with 0.4 bpp. We find this process will improves the security performance by about 1.0%. For 0.2 bpp, we only compare with S-UNIWARD because work [22] did not run experiment on this payload. It is observed from Table IV that the proposed UT-GAN performs better than ASDL-GAN by 4.96% and 7.80% for 0.4 bpp and 0.1 bpp respectively. It also obtained better performance than S-UNIWARD. From Table V, the incorporation of SCA can also improve the performance on full size image of 512×512 . The performance improvement of incorporation of SCA will be more significantly with the increment of payload. This is may because of the hard training for low payload by using deep-leaning based method, no matter steganalysis or steganography.

TABLE IV: Error rates (%) of different steganographic schemes detected by SRM [14] on BOSSBase 512×512 .

Payload	UT-GAN	ASDL-GAN [22]	S-UNIWARD [5]
0.4 bpp	22.36	17.40	20.54
0.2 bpp	33.03	/	31.89
0.1 bpp	40.82	33.02	40.40

TABLE V: Error rates (%) of different steganographic schemes detected by maxSRMd2 [28].

Payload	UT-SCA-GAN	UT-GAN
0.4 bpp	20.42	18.23
0.2 bpp	28.04	26.87
0.1 bpp	34.89	34.64

In addition, we also compare the training time of UT-GAN and ASDL-GAN in one epoch. ASDL-GAN needs 4.65 hours, while UT-GAN only needs 1.30 hours. Thus the proposed method saves almost 100 hours for 32 epoch (41.6 hours vs 148.8 hours). There are two reasons for this difference. One reason is the simple architecture of proposed generator, as

opposed to the 25-layer generator of the ASDL-GAN. What's more, the time consumption of the proposed Tanh-Simulator function is negligible compared to the two independent 4-layer TES sub-network of ASDL-GAN.

We present the embedding change probability maps and modification position maps with payloads of 0.4 bpp and 0.1 bpp in Fig. 5. From (b) and (d), it can be seen that embedding change probability value of the texture regions are larger than the smooth regions. From (c) and (e), the embedding change position are also concentrated on regions with large embedding change probability values. Fig. 5 shows that the proposed steganography scheme is content-adaptively.

IV. PRACTICAL APPLICATION OF THE PROPOSED ARCHITECTURE

In this work, our task is to learn the embedding probability $p_{i,j}$ by adversarial training. For every pixel of the cover image, the proposed Tanh-simulator has been used to simulate the embedding process. In a practical application scenario, it is necessary to compute the embedding cost by taking full use the learned embedding probability, and then using the Syndrome Trellis codes [1] to embed the secret information. The embedding cost ρ_{ij} for the practical steganographic coding scheme can be computed as follows [6]:

$$\rho_{ij} = \ln(1/p_{i,j} - 2)) \quad (10)$$

We embed the binary image that has only two possible values 0 and 1 into the cover image. The flowchart of embedding and extracting are show on Fig. 6. The embedding change probability denotes the probability learned form adversarial training. Fig. 7 show an example of embedding process. Experimental results show that all of the secret message can be recovered by STC scheme, and the embedding positions are also located in complex region. Thus the proposed steganography scheme can improve the security performance and also can be used for practical application.

V. CONCLUSION

In this paper, a secure steganographic scheme based on generative adversarial network is proposed. A Tanh-simulator function was proposed to fit the optimal embedding simulator. A compact generator based on U-Net architecture is employed as the generator. To resist the current advanced steganalysis methods maxSRMd2, selection channel awareness (SCA) are incorporated into the discriminator. Experimental results showed that the proposed architecture outperforms the ASDL-GAN method dramatically by using less training time. Furthermore, it also obtained better performance than the hand-crafted method S-UNIWARD.

In our future work, we will explore the relationship between the architectures of generator and discriminator so as to boost the security performance. In addition, we would like to apply the proposed scheme to the JPEG domain.

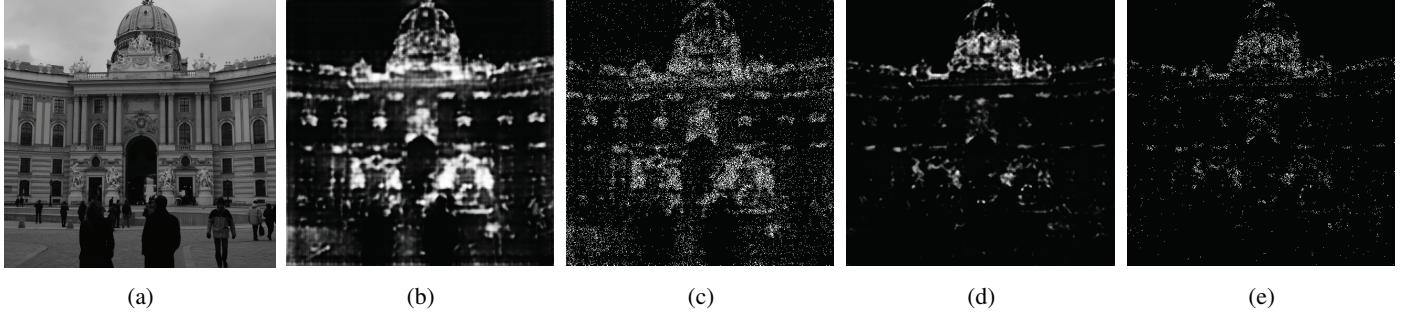


Fig. 5: Illustration of the proposed UT-GAN. (a) The BOSSBase cover image “1013.pgm” with a size of 512×512 , (b) embedding change probability (0.4 bpp), (c) modification map (0.4 bpp), (d) embedding change probability map (0.1 bpp), (e) modification map (0.1 bpp).

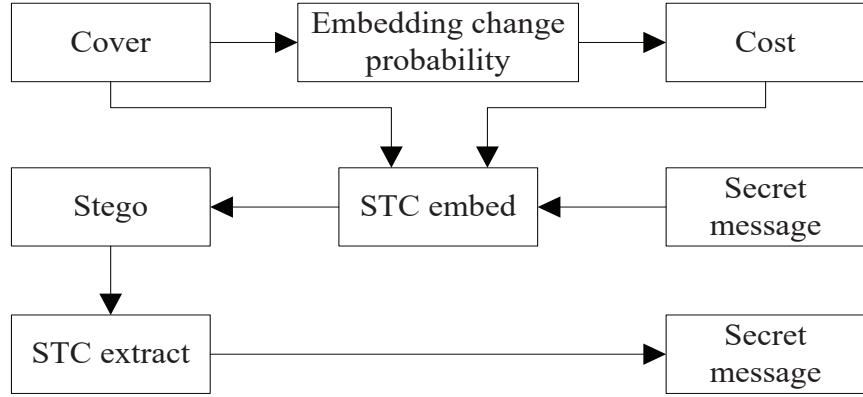


Fig. 6: Flowchart of practical embedding process.



Fig. 7: Illustration of the practical application of proposed UT-SCA-GAN. (a) The BOSSBase cover image “1013.pgm” with a size of 256×256 , (b) secrete message (with a size of 128×128), (c) stego image, (d) modification map, (e) recovered message.

REFERENCES

- [1] Tomáš Filler, Jan Judas, and Jessica Fridrich. Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Transactions on Information Forensics and Security*, 6(3):920–935, 2011.
- [2] Tomáš Pevný, Tomáš Filler, and Patrick Bas. Using high-dimensional image models to perform highly undetectable steganography. In *International Workshop on Information Hiding*, pages 161–177. Springer, 2010.
- [3] Vojtech Holub and Jessica Fridrich. Designing steganographic distortion using directional filters. In *Information Forensics and Security (WIFS), 2012 IEEE International Workshop on*, pages 234–239. IEEE, 2012.
- [4] Bin Li, Ming Wang, Jiwu Huang, and Xiaolong Li. A new cost function for spatial image steganography. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 4206–4210. IEEE, 2014.
- [5] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark.

- Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014(1):1, 2014.
- [6] Vahid Sedighi, Rémi Cogranne, and Jessica Fridrich. Content-adaptive steganography by minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security*, 11(2):221–234, 2016.
 - [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
 - [8] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
 - [9] Shunquan Tan and Bin Li. Stacked convolutional auto-encoders for steganalysis of digital images. In *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA)*, pages 1–4. IEEE, 2014.
 - [10] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Deep learning for steganalysis via convolutional neural networks. *Media Watermarking, Security, and Forensics*, 9409:94090J–94090J, 2015.
 - [11] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Learning and transferring representations for image steganalysis using convolutional neural network. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 2752–2756. IEEE, 2016.
 - [12] Guanshuo Xu, Han-Zhou Wu, and Yun-Qing Shi. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, 23(5):708–712, 2016.
 - [13] Guanshuo Xu, Han-Zhou Wu, and Yun-Qing Shi . Ensemble of cnns for steganalysis: an empirical study. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pages 103–107. ACM, 2016.
 - [14] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.
 - [15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.
 - [16] Jianhua Yang, Kai Liu, Xiangui Kang, Edward Wong, and Yunqing Shi. Steganalysis based on awareness of selection-channel and deep learning. In *International Workshop on Digital Watermarking*, pages 263–272. Springer, 2017.
 - [17] Jian Ye, Jiangqun Ni, and Yang Yi. Deep learning hierarchical representations for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 12(11):2545–2557, 2017.
 - [18] Jianhua Yang, Yun-Qing Shi, Edward K Wong, and Xiangui Kang. Jpeg steganalysis based on densenet. *arXiv preprint arXiv:1711.09335*, 2017.
 - [19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
 - [20] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
 - [21] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint*, 2016.
 - [22] Weixuan Tang, Shunquan Tan, Bin Li, and Jiwu Huang. Automatic steganographic distortion learning using a generative adversarial network. *IEEE Signal Processing Letters*, 24(10):1547–1551, 2017.
 - [23] Jessica Fridrich and Tomas Filler. Practical methods for minimizing embedding impact in steganography. In *Electronic Imaging 2007*, pages 650502–650502. International Society for Optics and Photonics, 2007.
 - [24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.
 - [25] Tomáš Denemark, Jessica Fridrich, and Vojtěch Holub. Further study on the security of s-unoward. In *Media Watermarking, Security, and Forensics 2014*, volume 9028, page 902805. International Society for Optics and Photonics, 2014.
 - [26] Patrick Bas, Tomáš Filler, and Tomáš Pevný. ? break our steganographic system?: The ins and outs of organizing boss. In *Information Hiding*, pages 59–70. Springer, 2011.
 - [27] Jan Kodovsky, Jessica Fridrich, and Vojtěch Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, 7(2):432–444, 2012.
 - [28] Tomas Denemark, Vahid Sedighi, Vojtech Holub, Rémi Cogranne, and Jessica Fridrich. Selection-channel-aware rich model for steganalysis of digital images. In *Information Forensics and Security (WIFS), 2014 IEEE International Workshop on*, pages 48–53. IEEE, 2014.