Journal of Information Hiding and Multimedia Signal Processing ©2022 ISSN 2073-4212
Ubiquitous International Volume **13**, Number **4**, December **2022**

Incorporating Clustering Modification Directions into Reinforcement Learning Based Cost Learning Framework

Jihua Cui*

Harbin Electric Power Vocational Technology College
59 Xiangdian Street, Harbin, China
lhf_cjh@sina.com

Zhenbang Wang and Shigang Tian

State Grid Heilongjiang Electric Power Co.,Ltd.
301 Hanshui Road, Harbin, China
zhenbangw@163.com; 53688067@163.com

Junfeng Zhao and Shen Wang

Harbin Institute of Technology
92 West Da Zhi St., Harbin, China
junfengzhao@hit.edu.cn; shen.wang@hit.edu.cn

Received October 2022; revised November 2022

Abstract. *Content-adaptive image steganography embedding cost learning frameworks based on deep learning can generate a more exquisite embedding probability map within a short time, and such methods have reached remarkable security performance compared to conventional hand-craft based methods and received increasing attention in recent years. However, existing Reinforcement Learning (RL)-based schemes are typically based on single-step state machine, making it difficult for further improvement. This paper extends the existing RL-based framework into two steps to enhance the simulated stego images from policy network to improve the performance, that is, during the training process, similar to the conventional methods, a module will be added after the policy network, the current embedding direction is adjusted according to the sign of modification directions of the neighborhood. The experimental results show that the proposed module not only improve the performance during the training process, but also enhance the actual security performance compared with single-step based frameworks when countering multiple steganalyzers.*
Keywords: Steganography, Steganalysis, Reinforcement learning, Content adaptive

1. **Introduction.** Image steganography attempts to embed message bits into public cover images in an invisible way, and the goal it pursues is to prevent the attackers from discovering the stego images. But earlier steganography methods only emphasize on embedding capacity, not embedding quality, which can be broken rapidly by conventional statistical based methods [1].

Until now, content-adaptive steganographic schemes under the framework of minimizing distortion function are more efficient than the conventional methods, which are the mainstream in modern steganography. This kind of schemes calculate the embedding costs of image to evaluate where to embed best, then get modification map by embedding

simulator [2] or STC [3] encoder, finally add to cover image to obtain a stego image, and this idea has been utilized widely in HUGO [4], WOW [5], S-UNIWARD [6], HILL [7].

In contrast, steganalysis is a kind of methods to attack steganography, which is used to detect whether the suspicious image is stego or not. Early steganalysis methods detect images mainly by using statistical based methods. In order to further improve the detection accuracy, the emergence of hand-craft based steganalysis methods altered the situation to get the relationship between neighbor pixels in different directions by using feature extraction, such as Spatial Rich Model (SRM[8]), then these features are fed into Ensemble Classifier(EC[9]) to detect images.

In recent years, benefit from optimizing richer features in deep learning, the design of CNN-based steganalysis schemes captures the attention of researchers. For CNN-based steganalysis on small images, GNCNN [10] was proposed by Qian et al, which consists of a preprocessing module based on KV high-pass filter, and its performance has been closed to SRM, which laid the foundation for follower research. An improved version of GNCNN called IGNCNN[11], which presents the ability of combining the trained CDNs in a multimodal framework and studies the effect of this combination on the detection accuracy. Xu et al.[12] proposed Xu-Net, which uses absolute and tanh functions to enhance the preprocessing results, and multiple structural groups are designed to adjust them thus improving performance. Yang et al. [13] proposed maxCNN, which first combined selection channel with CNN steganalysis. Ye et al. [14] proposed Ye-Net, all SRM high-pass filters and selection channel are utilized as preprocessing, Truncated Linear Unit (TLU) function was designed to further enhance the results from preprocessing layers, finally achieved competitive performance. On this basis, Yedroudj et al.[15] proposed an improved version called Yedroudj-Net, optimizing their model by simultaneously using linear and non-linear filters. To further improve performance. Borouman et al. [16] proposed SR-Net, which initializes filters randomly and extracts residual through multiple non-pooling convolutional layers. You et al.[17] proposed a Siamese CNN to solve the problem of staganalysis of the images in arbitrary size, which consists of two symmetrical subnets with shared parameters, thus making their model well-generalized and robust.

Although researchers have made in-depth studies in CNN-based steganalysis, which have posed great challenge to image steganography, In the past few years, GAN-based[18] or RL-based content-adaptive embedding frameworks present a booming development, and the security performance of those methods are excellent compared to conventional hand-craft based methods.

Tang et al.[19] first proposed a GAN-based steganography scheme ASDL-GAN(Automatic Steganographic Distortion Learning with Generative Adversarial Network). Similar to the process of hand-craft based methods, the generator is used to calculate embedding probability map of cover images, consisting of 25 groups with shortcut connections to identify the feature map of stack layers. According to the probability map, Ternary Embedding Simulator (TES) network is employed to simulate the process of embedding secret information. The performance of ASDL-GAN is close to the hand-craft based methods. However, TES network needs pre-training before using in GAN, which may lead to a long time of training, and the training effect may affect the final performance. Yang et al. [20] proposed UT-GAN (U-Net [21] and Tanh Embedding Framework Using GAN), which is an improved version of ASDL-GAN. U-Net based generator greatly reduces the number of model parameters. Two tanh-based functions are combined as embedding simulator to avoid the pre-training problem of TES, thus saving much time on training. Xu-Net with 6 HPF kernels is used as discriminator. UT-GAN has outperformed the conventional hand-craft based steganography schemes. The latest works adopt RL to alleviate the vanishing gradient problem, In SPAR-RL [22] (Steganographic Pixel-wise Actions and

Rewards with RL), a policy network attempted to learn embedding policy and decomposed the embedding into pixel-wise actions to maximize the rewards, a sampling process was designed to simulate the embedding actions and the gradients of data embedding were assigned to the reward function. As for some of the new technologies about information hiding, in reversible data hiding, a new method (HistC) [23] is proposed using histogram shifting on blocks, in which selected peak points are pixels at the center of the histogram distribution, and a new approach [24] is designed to conceal secret bits within Arabic text cover media via using Kashida and Unicode space characters (UniSpaCh), which are used in Microsoft word documents.

The security performance of the related works are relatively high. However, there are some limitations in those methods. However, the single-step RL framework cannot adjust the modification direction of the current position according to the situation of the neighborhood, which makes it difficult to further improve the training efficiency and performance. Inspired by CMD [25], this paper extends the existing RL framework into two steps to enhance the simulated steganography images from generators to solve this problem. That is, in the training process, the embedding simulator is used to simulate the embedding, and then the current embedding direction is adjusted according to the sign of modification directions of the neighborhood, and then the discriminator is fed by the enhanced simulated stego images to continue to complete the model training. In this way, the preprocessing module in the discriminator can be interfered to further improve the training effect of the generator.

2. **RL-based Embedding Cost Learning Framework.** The overall architecture of RL-based embedding cost learning framework is illustrated in Fig. 1. The policy network **P** learns embedding probability of pixel value $x_{i,j}$ according to the input images. In order to generate simulated stego images and guarantee stability of the framework, optimal embedding simulator is employed to deal with the data embedding, and a reward function **R** is designed to guarantee the gradients of data embedding. Similar to the discriminator of GAN-based frameworks, the environment network **E** can be referred as Xu-Net [12] with 6-HPF kernels, puts the cover images and the simulated stego images generated from **P** for training. Finally, calculate the loss of **P**, **E** and **R** separately and update the whole model through back-propagation. These operations of existing single-step RL-based framework can be treated as a state machine, which is shown as follows.

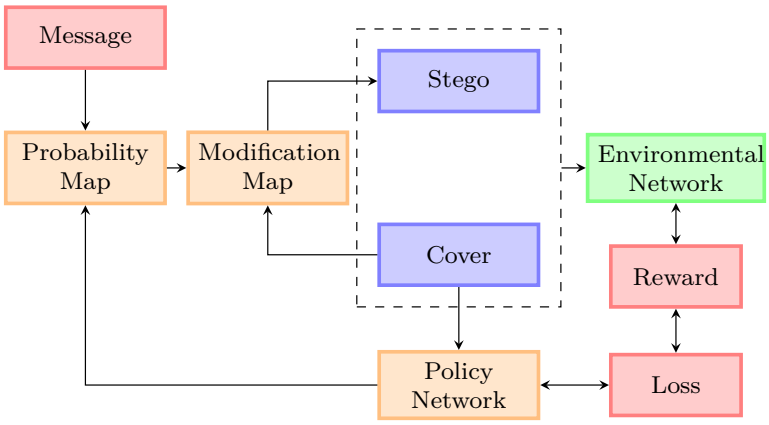


FIGURE 1. The overall architecture of single-step RL-based cost learning framework.

In Fig. 2, where s_n indicates the state of the n-th iteration, a_n, $R(a_n, s_n)$ is the corresponding action and reward of s_n. Note that the design of each components, such as

$$s_0 \xrightarrow[R(a_0,s_0)]{a_0} s_1 \xrightarrow[R(a_1,s_1)]{a_1} s_2 \cdots\cdots s_{T-1} \xrightarrow[R(a_{T-1},s_{T-1})]{a_{T-1}} s_T$$

FIGURE 2. State machine of single-step RL.

the functions of policy network, environment network, and reward in this article, can be referred in SPAR-RL [22], we will extend the existing work into multi-step.

3. **The Proposed Method Incorporating Clustering Modification Directions.** In fact, under the existing single-step RL-based framework, the embedding probabilities are learned by the policy network during training processes, but existing framework does not fully consider the impact of the neighborhood pixels. However, when enhancing the RL-based frameworks, simply adding a module (such as selection channel) to the environment network will strengthen its adversarial effect, eventually affect the actual performance of policy network. Therefore, it is necessary to further improve the policy network to enhance its segmentation capability, and the existing state machine from single-step to multi-step will be extended by investigating the modification directions of the simulated stego images during the training process and optimize them to improve the security performance.

Reviewing CMD [25], the main idea is to divide the image into multiple sub-blocks. When adjusting the modification direction of the next sub-block, the adjustment will be based on the neighborhood modifications of each element from the previous sub-blocks, and then accomplish the data embedding of the message bits of current sub-block until the data embedding is completed. However, due to the operations are typically matrix level in deep learning, simply applying this method may reduce the training may reduce the training efficiency of the model.

In order to solve this problem, this paper proposes a simplified strategy, where the goal is to interfere with the ability of extracting residuals of the preprocessing module in the environmental network. That is, accomplishing the embedding simulation in the training process first, and then adjust the modification directions of the neighborhood of each pixel to obtain the simulated stego images. As shown in Fig. 3, assume a pixel value $x_{i,j}$ at (i, j) with four neighborhoods in the horizontal and vertical directions, and following adjustments will be made according to the modification of the neighborhood.

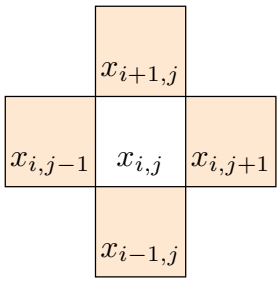


FIGURE 3. The neighborhood of $x_{i,j}$ in vertical and horizontal directions.

$$p^{+}_{i,j} = \begin{cases} p_{i,j} * \alpha, & \text{if } \sum_{(m,n)\in N_{i,j}} d_{m,n} > 0 \\ p_{i,j}, & \text{otherwise} \end{cases} \tag{1}$$

$$p^{-}_{i,j} = \begin{cases} p_{i,j} * \alpha, & \text{if } \sum_{(m,n)\in N_{i,j}} d_{m,n} < 0 \\ p_{i,j}, & \text{otherwise} \end{cases} \tag{2}$$

where α is the scaling factor, $p^{+}_{i,j}$, $p^{-}_{i,j}$ are the embedding probability of $+1$ and -1 respectively, $N_{i,j}$ is the neighbor of $x_{i,j}$, such as $x_{i+1,j}$, $x_{i-1,j}$, $x_{i,j+1}$, $x_{i,j-1}$, $d_{i,j}$ is the corresponding modification of $x_{i,j}$, typically is -1, 0, +1. Moreover, the embedding probabilities adjustment can be summarized as two following cases.

1. When the number of +1 modifications is greater than the number of -1 modifications, the modification direction of $x_{i,j}$ should be adjusted to -1, and the corresponding embedding probability should also be adjusted to ensure the modification directions in the continuous region (Fig. 3) are different.
2. When the number of -1 modifications is greater than the number of +1 modifications, the modification direction of $x_{i,j}$ should be similarly adjusted to +1, and the embedding cost should also be adjusted according to the above rules.

After the embedding probabilities adjustment in this way, the modification directions of the simulated stego images in each continuous area will not be synchronized during the training process, so as to affect interference environment network and strengthen the learning ability of the strategy network module on the image neighborhood.

In practical application, the proposed module will be placed after the embedded simulator, and its state machine will be adjusted from single step (Fig. 2) to multi-step (Fig. 4). For example, $s_{0,0}$ is the initial state, and $a_{0,1}$ is the conventional embedding simulation, $R(a_{0,1}, s_{0,1})$ is used to adjust the embedding probabilities according to the proposed method for the simulation from $a_{0,1}$ and turn to the $s_{0,1}$. After the probabilities adjustment, the step $a_{0,1}$ and $R(a_{0,1}, s_{0,1})$ is same as a_0 and $R(a_0, s_0)$ in Fig. 2 of single-step framework.

$$s_{0,0} \xrightarrow[R(a_{0,0},s_{0,0})]{a_{0,0}} s_{0,1} \xrightarrow[R(a_{0,1},s_{0,1})]{a_{0,1}} s_{1,0} \xrightarrow[R(a_{1,0},s_{0,0})]{a_{1,0}} s_{1,1} \xrightarrow[R(a_{1,1},s_{0,1})]{a_{1,1}} s_{2,0} \cdots\cdots$$

FIGURE 4. The two-step state machine of the RL-based framework, each state contains two sub-states, and where the first reward is used to represent the proposed method, the second reward is the conventional reward function based on single-step version.

To this end, the practical implementation steps are as follows.

Step 1 Use the policy network to get the original embedding probabilities and generate the modification map for the embedding simulation.

Step 2 According to the modification map of **Step 1**, adjust the elements at the matrix level in the neighborhood as shown in Fig. 3 based on the proposed method and obtain the adjusted embedding probabilities. The conceptual diagram of the adjustment is shown in Eq, 1 and Eq. 2.

Step 3 Generate a new simulation modification map with new embedding probabilities.

Step 4 Add the new simulation modification map to the original image to get the final simulated stego image, and put them together into the environmental network to accomplish the training iteration.

Step 5 Get the Loss and Reward for backpropagation, and turn to **Step 1** to the next iteration.

4. **Experiments.**

4.1. **Experimental Setup.**

The settings of our experiments are as follows, ALASKA#2, BOSSBase v1.01[26] and BOWS2[27] datasets are used to train the GAN or RL-based frameworks or evaluate the security performance in different cases, containing 10,000 gray-scale images with 512×512 resolution separately. To improve the efficiency of experiments

and avoid wasting time, all of the original images were resized to 256×256. We use Adam optimizer and set the learning rate to 0.0001, and half of the images of ALASKA#2 will be used on the training phase, each batch contains 24 cover images were used to generate the corresponding stego images. BOSSBase v1.01 and BOWS2 were used to generate the final simulated stego images with the trained model. The parameter settings of the existing modules of the framework were same as SPAR-RL [22], and α in Eq. 1 and Eq. 2 is set to 2, and it was performed in Tensorflow v1.15, all of the experiments are conducted on with Intel XEON 4216 CPU and NVIDIA GTX 1080Ti graphics card.

In order to evaluate the effectiveness of the proposed module, multiple steganography schemes are adopted, which is combined in the following ways:

- SPAR-RL-v2: RL-based structural design, whose generator and discriminator were from UT-GAN [20]. It contains U-Net based generator and Xu-Net with 6-HPF kernels. Optimal embedding simulator is employed to maintain the stability, and a reward function is designed to guarantee the gradient of the embedding simulation during the training process.
- SPAR-RL-v2-multi: An improved version of SPAR-RL-v2 by adding the proposed module after the policy network in SPAR-RL-v2, and the state machine of the model will be multi-step.

In our experiments, three state-of-the-art steganalysis schemes were selected to evaluate security performance, including one hand-crafted based and two state-of-the-art CNN-based steganalysis schemes are as follows:

- SRM [8]: Multiple high-pass filters are used to obtain image residuals, 4th-order co-occurrence matrix is extracted from quantized and truncated residuals, and FLD-based ensemble classifier is used for classification.
- Xu-Net [12]: Similar to the discriminator of UT-GAN, 6-HPF kernels was utilized as preprocessing layer, and 5 groups of convolutional layers are used on feature extraction. Absolute and tanh activation functions are used to limit the scope after preprocessing. 1×1 convolution, global pooling and BN layer are added to improve detection performance.
- Yedroudj-Net [15]: 30-HPF kernels are employed for boosting detection performance to obtain different kinds of residuals. TLU activation function is also applied in feature maps to prevent modeling outlier values in deep layers.

These steganalysis schemes are used to evaluate security performance of above steganography schemes on BOSSBase v1.01 and BOWS2 dataset. For hand-craft based methods SRM, the training set and test set are divided in 1:1. For the above two CNN-based steganalysis model, the training set, verification set and test set are divided into into 5:1:4. In our experiment, detection error rate is a way to measure the security performance, is defined as follows:

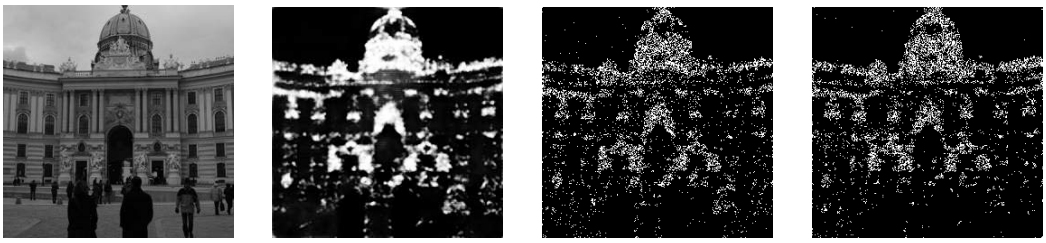
$$P_{\mathrm{E}} = \min_{P_{\mathrm{FA}}} \frac{1}{2} \left(P_{\mathrm{FA}} + P_{\mathrm{MD}} \left(P_{\mathrm{FA}} \right) \right) \tag{3}$$

where P_{FA} is false alarm rate, P_{MD} is detection error rate.

4.2. **Security Performance.** In this subsection, the evaluation of security performance on these steganography schemes has been imple- mented, models are divided into multiple groups to verify the improvement of our framework by each sub-module. The security performance based on SRM and CNN-based steganalyzers are shown in Table 1. During our evaluation, each scheme is trained at 0.4 bpp payload without curriculum learning, and fine-tuning are adopted to adjust the models in different payload to reach the best state.

TABLE 1. Performance Comparison Among Different Steganographic Schemes

Steganalyzer	Steganography	Payload	
		0.2bpp	0.4bpp
SRM	SPAR-RL-v2	0.3842	0.2732
	SPAR-RL-v2-multi	0.3910	0.2781
Xu-Net	SPAR-RL-v2	0.4169	0.3342
	SPAR-RL-v2-multi	0.4208	0.3395
Yedroudj-Net	SPAR-RL-v2	0.3725	0.3024
	SPAR-RL-v2-multi	0.3767	0.3055



(A) (B) (C) (D)

FIGURE 5. Comparison of modification map between different steganographic schemes in "1013.pgm" of BOSSBase v1.01. (a) Original image. (b) Embedding probability map with the wide U-shape based policy network [20]. (c),(d): modification map generated under different schemes.

As we can see from Table 1, the proposed module in this paper are effective to against CNN-based steganalysis models by adding one step to the state machine of the existing RL-based framework. Owing to the adjustment of the modification directions according to Sec. 3, we can get the modification map with less noise, and the visualization of embedding probability map and the modification map under is shown in Fig. 5.

4.3. **Comparison of training process.** In this section, the training process of the two schemes is compared to illustrate the effectiveness of the proposed module for the model trainings.

Fig. 6 visualizes the detection error rate of the two schemes during the training of simulation of the stego images. As seen, after 10000 iterations, SPAR-RL-v2-multi achieves higher detection accuracy than SPAR-RL-v2.

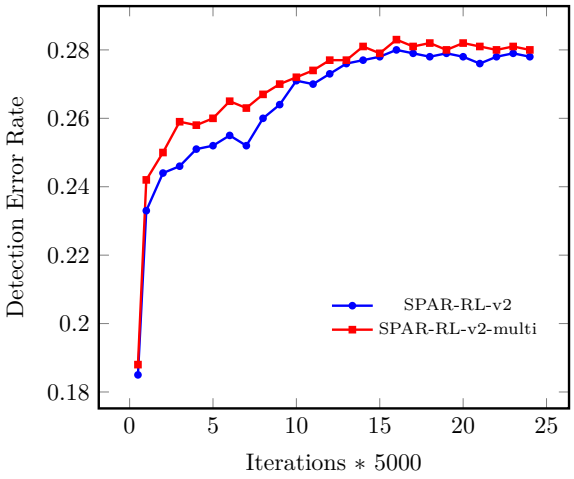


FIGURE 6. Comparison of training process.

According to the results, it is clear to observe that the use of the proposed module is somewhat more effective than the single-step method in the training. When training with the proposed module, with the help of the embedding probabilities adjustment, each region (like Fig. 3) of the modification map will not be synchronized. It can be seen that the proposed multi-step module achieves the better effect, which makes the model training more effective.

5. **Conclusions.** Content-adaptive embedding cost learning framework based on RL is crucial to the modern steganography. In this paper, we propose a module to extend existing single-step framework to multi-step to improve the training efficiency, so that the trained policy network can get better segmentation ability. Our results show that equipping the proposed module can improve the security performance at actual application level.

REFERENCES

[1] Andreas Westfeld and Andreas Pfitzmann. Attacks on steganographic systems. In *Information Hiding*, pages 61–76, 2000.

[2] Jessica Fridrich and Tomas Filler. Practical methods for minimizing embedding impact in steganography. In *Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 13–27, San Jose, CA, United States, 2007.

[3] 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.

[4] Tomáš Pevný, Tomáš Filler, and Patrick Bas. Using high-dimensional image models to perform highly undetectable steganography. In *Information Hiding*, pages 161–177, 2010.

[5] Vojtěch Holub and Jessica Fridrich. Designing steganographic distortion using directional filters. In *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 234–239, 2012.

[6] V. Holub, J. Fridrich, and Tomá Denemark. Universal distortion function for steganography in an arbitrary domain. *Eurasip Journal on Information Security*, (1):1, 2014.

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

[8] Jessica Fridrich and Jan Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012.

[9] 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.

[10] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Learning and transferring representations for image steganalysis using convolutional neural network. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2752–2756, 2016.

[11] Elshafey. Mohamed A, Amein Ahmed S, and Badran Khaled S. Universal image steganography detection using multimodal deep learning framework. *Journal of Information Hiding and Multimedia Signal Processing*, 12(3):152–161, 2021.

[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] Jianhua Yang, Kai Liu, Xiangui Kang, Edward Wong, and Yunqing Shi. Steganalysis based on awareness of selection-channel and deep learning. In *Digital Forensics and Watermarking*, 2017.

[14] 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.

[15] M. Yedroudj, F Comby, and M Chaumont. Yedroudj-net: An efficient cnn for spatial steganalysis. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2092–2096, 2018.

[16] M. Boroumand, M. Chen, and J. Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, 2019.

[17] W. You, H. Zhang, and X. Zhao. A siamese cnn for image steganalysis. *IEEE Transactions on Information Forensics and Security*, 16:291–306, 2021.

[18] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, 2014.

[19] W. Tang, S. Tan, B. Li, and J. Huang. Automatic steganographic distortion learning using a generative adversarial network. *IEEE Signal Processing Letters*, 24(10):1547–1551, 2017.

[20] J. Yang, D. Ruan, J. Huang, X. Kang, and Y. Shi. An embedding cost learning framework using gan. *IEEE Transactions on Information Forensics and Security*, 15:839–851, 2020.

[21] O Ronneberger, P Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, 2015.

[22] Weixuan Tang, Bin Li, Mauro Barni, Jin Li, and Jiwu Huang. An automatic cost learning framework for image steganography using deep reinforcement learning. *IEEE Transactions on Information Forensics and Security*, 16:952–967, 2021.

[23] Nguyen Kim Sao, Cao Thi Luyen, and Pham Van At. Efficient reversible data hiding using block histogram shifting with invariant peak points. *Journal of Information Hiding and Multimedia Signal Processing*, 13(1):78–97, 2022.

[24] Shareef Taka Farah R. Text steganography based on noorani and darkness. *Journal of Information Hiding and Multimedia Signal Processing*, 12(3):127–139, 2021.

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

[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, 2011.

[27] Bas Patrick and Furon Teddy. Bows-2, 2007.