CNN-Based Adversarial Embedding for Image Steganography

Weixuan Tang, *Student Member, IEEE*, Bin Li, *Senior Member, IEEE*, Shunquan Tan, *Senior Member, IEEE*, Mauro Barni, *Fellow, IEEE*, and Jiwu Huang, *Fellow, IEEE*

***Abstract*— Steganographic schemes are commonly designed in a way to preserve image statistics or steganalytic features. Since most of the state-of-the-art steganalytic methods employ a machine learning (ML)-based classifier, it is reasonable to consider countering steganalysis by trying to fool the ML classifiers. However, simply applying perturbations on stego images as adversarial examples may lead to the failure of data extraction and introduce unexpected artifacts detectable by other classifiers. In this paper, we present a steganographic scheme with a novel operation called adversarial embedding (ADV-EMB), which achieves the goal of hiding a stego message while at the same time fooling a convolutional neural network (CNN)-based steganalyzer. The proposed method works under the conventional framework of distortion minimization. In particular, ADV-EMB adjusts the costs of image elements modifications according to the gradients back propagated from the target CNN steganalyzer. Therefore, modification direction has a higher probability to be the same as the inverse sign of the gradient. In this way, the so-called adversarial stego images are generated. Experiments demonstrate that the proposed steganographic scheme achieves better security performance against the target adversary-unaware steganalyzer by increasing its missed detection rate. In addition, it deteriorates the performance of other adversary-aware steganalyzers, opening the way to a new class of modern steganographic schemes capable of overcoming powerful CNN-based steganalysis.**

***Index Terms*— Steganography, steganalysis, adversarial machine learning.**

I. Introduction

IMAGE steganography is the art and science of concealing covert information within images. It is usually achieved by modifying image elements, such as pixels or DCT coefficients. On the other side of the game, steganalysis aims to reveal the presence of secret information by detecting whether there are abnormal artifacts left by data embedding.

The developing history of steganography and steganalysis is rich of interesting stories, as they compete with each other and they benefit and evolve from the competition [1]. The earliest steganographic method was implemented by substituting the least significant bits of image elements with message bits. The stego artifacts introduced by this method can be effectively detected by Chi-squared attack [2], or steganalytic features based on *first-order statistics* [3]. In this initial phase of the competition, statistical hypothesis testing or a simple linear classifier such as FLD (Fisher Linear Discriminant) can serve the need of steganalysis. The first-order statistics can be restored after data embedding, as was done in [4]. The abnormal artifacts in the first-order statistics can also be avoided as in [5] and [6]. As a consequence, more powerful steganalytic features based on the *second-order statistics* [7], [8] were proposed. In this period, advanced machine learning (ML) tools, such as SVM (Support Vector Machine), were operated on high-dimensional features (where the dimension is typically several hundreds). These methods were very effective in detecting steganographic schemes even if the first-order statistics were preserved. Modern steganographic schemes are designed under the framework of *distortion minimization* [9]. The embedding cost of changing each image element is specified by a cost function, and a coding scheme is employed to convey information by minimizing the distortion, which is computed as the total cost of modified elements. The schemes in [10]–[15] use effective cost functions. As a counter-measure, state-of-the-art steganalytic methods adopt *higher-order statistics* with much higher dimensional features (where the dimension is typically thousands or even more than ten thousands), such as in [16]–[20]. More sophisticated ML methods, such as the ensemble classifier [21], have also been employed. Steganalytic methods based on deep learning [22]–[27] have rapidly gained an increasing attention in recent years. Without the need of designing hand-crafted features, deep convolutional neural networks (CNN) shows a promising way in automatic feature extraction and classification for steganalysis. Incorporating some prior domain knowledge into the network design, such as using high-pass filters for preprocessing, outstanding performance can be obtained.

The high-dimensional hand-crafted or deep-learned features with the powerful supervised ML schemes present a great challenge to steganography. A promising strategy for

Manuscript received March 11, 2018; revised July 20, 2018 and October 12, 2018; accepted December 26, 2018. Date of publication January 7, 2019; date of current version May 9, 2019. This work was supported in part by NSFC under Grants U1636202, 61572329, 61872244, and 61772349, in part by the Shenzhen Research Development Program under Grant JCYJ20160328144421330, and in part by the Alibaba Group through the Alibaba Innovative Research (AIR) Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Tomas Pevny. *(Corresponding author: Bin Li.)*

W. Tang is with the School of Electronics and Information Technology, Sun Yat-sen University, Guangdong 510006, China, and also with the Guangdong Key Laboratory of Intelligent Information Processing and Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen 518060, China (e-mail: tweix@mail2.sysu.edu.cn).

B. Li, S. Tan, and J. Huang are with the Guangdong Key Laboratory of Intelligent Information Processing and Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen 518060, China, and also with the National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen 518060, China (e-mail: libin@szu.edu.cn; tansq@szu.edu.cn; jwhuang@szu.edu.cn).

M. Barni is with the Department of Information Engineering and Mathematics, University of Siena, 53100 Siena, Italy (e-mail: barni@dii.unisi.it).

Digital Object Identifier 10.1109/TIFS.2019.2891237

the steganographer is to use side information which is not available to the steganalyst, such as using the camera sensor noise during message embedding [28] and the compression noise during JPEG compression [12]. However, the side information is not always available for all kinds of cover images, especially for those already compressed in JPEG format. As a consequence, better steganographic schemes suitable for more general conditions are needed.

As the dimension of steganalytic features increases, it is difficult for steganograhpy to preserve all statistical features during data embedding. This motivates us to find a better way to resist steganalysis by countering the ML based classifier. Recent studies [29], [30] have shown that ML systems are vulnerable to intentional adversarial operations. For example, Do *et al.* [31] showed that the image retrieval system based on SIFT features could be attacked by tweaking the keypoint orientations. Chen *et al.* [32] showed that the performance of an image forensics detector with a SVM classifier could be greatly degraded by a rather simple gradient based attack. There are also some research evidences indicating that classifier based on deep learning can be easily fooled by *adversarial examples* [33]–[37], which are formed by applying small but intentional perturbations to inputs in order to make the classification model yield erroneous outputs. However, applying adversarial perturbations as in [33] on stego images may lead to data extraction failures. The perturbations may also introduce unexpected artifacts detectable by other classifiers.

The progress in adversarial signal processing [38] inspired us to design a steganographic scheme that is resistant against ML based steganalyzers. In this paper, we propose a scheme called ADV-EMB (Adversarial Embedding). Targeted to counter a deep learning based steganalyzer [26], we generate stego images via *adversarial embedding*, an operation that takes into account both the embedding of the stego message and the necessity to fool the target steganalyzer. ADV-EMB is implemented under the framework of distortion minimization, and based on a baseline steganographic scheme adopting a conventional embedding mechanism. Specifically, ADV-EMB adapts the cost assignment process by asymmetrically adjusting a portion of embedding costs according to the gradients backpropagated from the deep learning steganalyzer. In order to avoid unnecessary extra modifications, the amount of image elements with adjustable costs is kept to be minimal. Experimental results show that the *adversarial stego images* generated by ADV-EMB can successfully fool the target deep learning steganalyzer, which was trained with several hundreds of thousands of training images.

Note that although to some extent being similar to the ASO (Adaptive Steganography by Oracle) scheme [39], [40] which also utilizes the information of a classifier, ADV-EMB does not aim to preserve any specific statistical model and does not directly generate embedding costs as [39] and [40]. ADV-EMB has a wider range of application than SI-UNWIARD [12], as it does not require side information which is only available at the steganographer's side. The target steganalyzer can be constructed on the steganographer's side and does not need to be exactly the same as the one used by a steganalyst. At the same time as the submission of this article, a related work proposed by Zhang *et al.* [41] tried to iteratively add adversarial perturbations on cover images first, and then embed messages into the "enhanced" cover images. The stego images generated in this way are robust against the detection of the target steganalyzer. However, the perturbations may introduce unexpected artifacts detectable by other non-target steganalyzers. In contrast, by our proposed method without over-adaption, although the adversarial stego images have a slightly higher rate of modifications then conventional stego images, they are less detectable by other advanced hand-crafted feature based steganalyzers and deep learning based steganalyzers.

The main contributions of our work are as follows:

1) A new strategy to fool the ML based steganalyzers, which is not based on the attempt to preserve a specific image statistical model, is proposed. We believe this is a promising way to counter steganalysis.
2) A practical steganographic scheme called ADV-EMB with adversarial embedding operation is proposed. As opposed to conventional approaches used to create adversarial examples in other machine learning domain, adversarial stego images generated by the ADV-EMB scheme are capable of carrying secret information.
3) Based on the knowledge available to the steganographer and the steganalyst, different adversarial models are considered, wherein the proposed scheme can achieve state-of-the-art security performance.

The rest of the paper is organized as follows. In Section II, we give the foundation of the proposed steganographic scheme, and differentiate two kinds of adversarial scenarios. We present the idea as well as a practical implementation of the proposed ADV-EMB steganographic scheme in Section III. Extensive experiments are performed and the results are reported in Section IV to demonstrate the performance of the ADV-EMB scheme under different adversarial conditions when compared to a baseline steganographic method. Conclusions are presented in Section V.

II. Technical Foundation

In this article, capital letters in bold are used to represent matrices. The corresponding lowercase letters are used for matrix elements. The flourish letters are used for sets. Specifically, cover and stego images are respectively denoted as $\mathbf{C} = (c_{i,j})^{H\times W}$and $\mathbf{S} = (s_{i,j})^{H\times W}$, where H and W are the height and width of the image. We use $\mathbf{Z} = (z_{i,j})^{H\times W} \in \mathcal{Z}$ to denote the proposed *adversarial stego image*. Note that $\mathbf{Z}$ is a special type of $\mathbf{S}$. The corresponding image sets are denoted as $\mathcal{C}$, $\mathcal{S}$, and $\mathcal{Z}$, respectively.

A. Practical Evaluation Metrics for Steganographic Security

The fundamental requirement of steganalysis is to differentiate stego images from cover images. To accomplish this task in a supervised ML setting, for analyzing an image $\mathbf{X}$, the steganalyzer may train a classifier $\phi_{\mathcal{C},\mathcal{S}}$ with binary output using training data from $\mathcal{C}$ and $\mathcal{S}$, and obtain the decision criterion as follows:

$$\begin{cases} \mathbf{X} \text{ is a cover image}, & \text{if } \phi_{\mathcal{C},\mathcal{S}}(\mathbf{X}) = 0, \\ \mathbf{X} \text{ is a stego image}, & \text{if } \phi_{\mathcal{C},\mathcal{S}}(\mathbf{X}) = 1. \end{cases} \quad (1)$$

The trained classifier is called *steganalyzer*. The missed detection happens when stego images are misclassified, and the false alarm happens when cover images are misclassified. The corresponding error probabilities are defined as:

$$P_{md}^{\phi_{\mathcal{C},\mathcal{S}}} = \Pr\{\phi_{\mathcal{C},\mathcal{S}}(\mathbf{S}) = 0\}, \tag{2}$$

and

$$P_{fa}^{\phi_{\mathcal{C},\mathcal{S}}} = \Pr\{\phi_{\mathcal{C},\mathcal{S}}(\mathbf{C}) = 1\}. \tag{3}$$

Under equal Bayesian prior for cover and stego, the total error rate is

$$P_{e}^{\phi_{\mathcal{C},\mathcal{S}}} = \frac{P_{md}^{\phi_{\mathcal{C},\mathcal{S}}} + P_{fa}^{\phi_{\mathcal{C},\mathcal{S}}}}{2}. \tag{4}$$

The goal of a steganalyst is to minimize $P_{e}^{\phi_{\mathcal{C},\mathcal{S}}}$, while the goal of a steganographer is the opposite.

B. Distortion Minimization Framework for Steganography

Under the distortion minimization framework, steganography is formulated as an optimization problem with a payload constraint, *i.e.*,

$$\min_{\mathbf{S}} \; D(\mathbf{C}, \mathbf{S}), \quad \text{s.t. } \psi(\mathbf{S}) = k, \tag{5}$$

where $D(\mathbf{C}, \mathbf{S})$ is a function measuring the distortion caused by modifying $\mathbf{C}$ to $\mathbf{S}$, $\psi(\mathbf{S})$ represents the message payload extracted from $\mathbf{S}$, and k is the amount of payload (measured in bits). A typical additive distortion function for ternary embedding, such as those used in [11]–[15], is defined as:

$$D(\mathbf{C}, \mathbf{S}) = \sum_{i=1}^{H}\sum_{j=1}^{W} \rho_{i,j}^{+}\delta(m_{i,j} - 1) + \rho_{i,j}^{-}\delta(m_{i,j} + 1), \tag{6}$$

where $m_{i,j} = s_{i,j} - c_{i,j}$ is the difference between the cover and the stego elements, $\delta(\cdot)$ is an indication function:

$$\delta(x) = \begin{cases} 1, & x = 0, \\ 0, & otherwise, \end{cases} \tag{7}$$

and $\rho_{i,j}^{+}$ and $\rho_{i,j}^{-}$ are respectively the cost of increasing and decreasing $c_{i,j}$ by 1. Although different steganographic schemes may employ different cost functions, a rule of thumb is that large cost values are assigned to elements more likely to introduce abnormal artifacts and thus leading to low probabilities of modification, and vice versa. In most schemes, $\rho_{i,j}^{+} = \rho_{i,j}^{-}$, leading to equal probabilities of increasing or decreasing $c_{i,j}$. With the CMD (clustering modification direction) strategy [42], [43], the costs of increasing or decreasing are asymmetrically updated during embedding in favor to a synchronized direction in neighborhood.

C. Steganographer's Knowledge About Steganalyzer

The steganographer may have different levels of knowledge about $\phi_{\mathcal{C},\mathcal{S}}$, such as the classification scheme and the training data. In this paper, we will not discuss what is the best strategy the steganographer should take according to the accessibility of these knowledge. Instead, we assume the gradients of the loss function with respect to the input, which are backpropagated from a ML based steganalyzer $\phi_{\mathcal{C},\mathcal{S}}$, are accessible to the steganographer. This is the foundation of the proposed steganographic scheme. In Section III, we will propose a scheme to fool such a steganalyzer with *adversarial stego images*. We will also investigate in the experimental part how well the adversarial stego images perform under other advanced steganalyzers (*e.g.*, $\phi'_{\mathcal{C},\mathcal{S}}$) when the knowledge of these steganalyzers is unavailable.

D. Steganalyst's Knowledge About Adversarial Stego Images

If a steganalyst is unaware of the adversarial operation presented to his steganalyzer, he is called *adversary-unaware steganalyst*. Otherwise, he is called *adversary-aware steganalyst*. One of the possible reactions of an adversary-aware steganalyst is to re-train the classifier with adversarial stego samples to obtain a new steganalyzer $\phi_{\mathcal{C},\mathcal{Z}}$, or use other advanced steganalyzers (*e.g.*, $\phi'_{\mathcal{C},\mathcal{Z}}$) *unknown to the steganographer.* This may present two challenging cases for a steganographer and we will discuss these scenarios in the experiments.

III. The Proposed ADV-EMB Steganographic Scheme

In this section, we will propose a novel steganographic scheme, called ADV-EMB, to counter a target steganalyzer. First, we will outline the basic idea of the proposed scheme. Then we will discuss the two most important operations in the proposed scheme, *i.e.,* adversarial embedding and minimizing the amount of adjustable elements, in detail. Finally, we will give a practical implementation of ADV-EMB.

A. Basic Idea

In the proposed scheme, the image elements are randomly divided into two groups, *i.e.,* a common group containing common elements, and an adjustable group containing *adjustable elements*. Data embedding is performed in two phases. In the first phase, a portion of the stego message is embedded into the common group by using *a conventional baseline steganographic scheme*. In the second phase, the remaining part of the stego message is embedded into the adjustable group by using *the proposed adversarial embedding scheme*. Adjustable elements are modified in such a way that a target steganalyzer would output a wrong class label. We use a well-performed deep learning based steganalyzer, *i.e.,* Xu's CNN [26], as the target steganalyzer, since the gradient values of its loss function with respect to the input can be used to guide the modification of adjustable elements. Other steganalyzers possessing such a property may also be used. The details will be given in Section III-B. In order to prevent over-adaption to the target steganalyzer and enhance the security performance against other advanced steganalyzers, the number of adjustable elements is minimized, resulting in a minimization problem with constraints. The details will be given in Section III-C.

In adversarial ML, an attack with full knowledge of a ML classifier is called a *white-box* attack. When the model, parameters and training data of the target classifier are not known, the attack is referred to as a *black-box* attack [44].

In our case, we adopt a white-box assumption in designing the steganographic scheme, however, we also test the new scheme in black-box scenarios against feature-based and CNN-based steganalyzers other than the targeted one (see Section IV).

B. Adversarial Embedding

Denote y as the ground truth label of $\mathbf{X}$. In steganalysis, we have $y \in \{0, 1\}$, where 0 indicates a cover and 1 indicates a stego. Let $L(\mathbf{X}, y; \phi_{\mathcal{C},\mathcal{S}})$ be the loss function of a steganalyzer $\phi_{\mathcal{C},\mathcal{S}}$. For example, for a deep neural network steganalyzer, the binary decision could be given as

$$\phi_{\mathcal{C},\mathcal{S}}(\mathbf{X}) = \begin{cases} 0, & \text{if } F(\mathbf{X}) < 0.5, \\ 1, & \text{if } F(\mathbf{X}) \geq 0.5, \end{cases} \tag{8}$$

where $F(\mathbf{X}) \in [0, 1]$ is the network output indicating the probability that $\mathbf{X}$ is a stego. The corresponding loss function may be designed in a form of cross entropy as

$$L(\mathbf{X}, y; \phi_{\mathcal{C},\mathcal{S}}) = -y \log\big(F(\mathbf{X})\big) - (1-y)\log\big(1 - F(\mathbf{X})\big) \tag{9}$$

In [33]–[35], adversarial examples are generated to fool ML models by updating input elements $x_{i,j}$ according to the gradient of the loss function with respect to the input (abbreviated as *gradient* if it is not specified otherwise), *i.e.*, $\triangledown_{x_{i,j}} L(\mathbf{X}, \hat{y}; \phi_{\mathcal{C},\mathcal{S}})$, by using a target label $\hat{y}$. However, it is impossible to directly apply these methods for securing steganography. In fact, modifying the elements of a stego image may lead to the failure of data extraction thus contradicting the aim of steganography. This motivates us to design an embedding method with two objectives of equal importance: performing adversarial operation to combat steganalyzer $\phi_{\mathcal{C},\mathcal{S}}$ and performing data embedding to carry information. To this end, we propose a method that we will call *adversarial embedding* to generate adversarial stego images under the framework of steganographic distortion minimization [9].

In [33], it is observed that when a perturbation signal associated with a target label is added to the input, the updated input, called *adversarial example*, is usually misclassified into the target class by the ML classifier. The perturbation signal can be designed in various ways, including using the gradient of the loss function with respect to the input. Since adding a perturbation with the inverse sign of the gradient has an adversarial effect, the objective of the proposed adversarial embedding is to modify image elements in such a way that the sign of the modification tends to be in accordance with the inverse sign of the gradient. To achieve such an objective with a high probability, together with data embedding, we operate under the distortion minimization framework by making the embedding costs bear the following properties:

$$\begin{cases} \rho^+_{i,j} < \rho^-_{i,j}, & \text{if } -\triangledown_{x_{i,j}}\ L(\mathbf{X}, \hat{y}; \phi_{\mathcal{C},\mathcal{S}}) > 0, \\ \rho^+_{i,j} = \rho^-_{i,j}, & \text{if } -\triangledown_{x_{i,j}}\ L(\mathbf{X}, \hat{y}; \phi_{\mathcal{C},\mathcal{S}}) = 0, \\ \rho^+_{i,j} > \rho^-_{i,j}, & \text{if } -\triangledown_{x_{i,j}}\ L(\mathbf{X}, \hat{y}; \phi_{\mathcal{C},\mathcal{S}}) < 0. \end{cases} \tag{10}$$

Such costs yield asymmetric probabilities of increasing and decreasing the element $x_{i,j}$, if the gradient is not zero. In this way, data can be embedded into the image elements, and the direction of the modification has the effect of inducing the steganalyzer $\phi_{\mathcal{C},\mathcal{S}}$ to decide for the target label $\hat{y} = 0$. Please note that the adversarial embedding may lead to higher modification rates due to the asymmetric embedding costs.

C. Minimum Amount of Adjustable Elements

With adversarial embedding, the adversarial stego images may effectively evade steganalysis. However, since the costs of increasing and decreasing are asymmetric, it increases the number of changed image elements. The reason is that the maximum entropy can only be obtained when the image element has an equal probability of increasing and decreasing. With the payload constraint, asymmetric costs lead to a higher change rate when compared to symmetric costs. Although a higher change rate may not necessarily lead to a worse security performance, we would still like to minimize it by reducing the frequency of adversarial embedding. This is due to three facts. First, it is sufficient to fool the ML classifier by using only a part of the elements to perform the adversarial operation, as shown in [44]. In fact, it is even unnecessary to perform adversarial embedding to those stego images which are generated by conventional steganographic schemes but are already incorrectly classified by the target steganalyzer. Second, if all elements are used for adversarial embedding, the generated adversarial stego images may be overly adapted to the target steganalyzer and may possibly become more detectable by other advanced steganalyzers. We may minimize the amount of elements for adversarial embedding to prevent introducing other detectable artifacts that can be exploited by an adversary-aware steganalyzer. Third, when the change rate is minimized, the image quality should be preserved better.

We propose to divide image elements into two groups *i.e.*, a common group containing common elements for conventional steganographic embedding, and an adjustable group containing adjustable elements for adversarial embedding. The objective is that the amount of adjustable elements should be minimized while the target steganalyzer should output a wrong class label. Mathematically speaking, the problem is formulated as

$$\min\ \beta, \quad \text{s.t. } \phi_{\mathcal{C},\mathcal{S}}(\mathbf{Z}) = 0 \ \text{ and } \psi(\mathbf{Z}) = k, \tag{11}$$

where $\beta \in [0, 1]$ denotes the ratio of the amounts of adjustable elements to all image elements, and $\psi(\cdot)$ and k have the same definition as in Eq.(5). It is obvious that there is no explicit solution to such a problem. To solve it efficiently, the target steganalyzer is employed to numerically search for "just enough" amount of adjustable elements to satisfy the constraints in (11). The details will be described in the next subsection.

D. A Practical Implementation of ADV-EMB

In this part, we present a practical ADV-EMB steganographic scheme. Since JPEG images are widely used and pervasive on the Internet, we use them as cover. We use Xu-CNN [26] as the target steganalyzer and J-UNIWARD [12] as the baseline steganographic scheme for conventional data embedding. The target steganalyzer is a CNN model

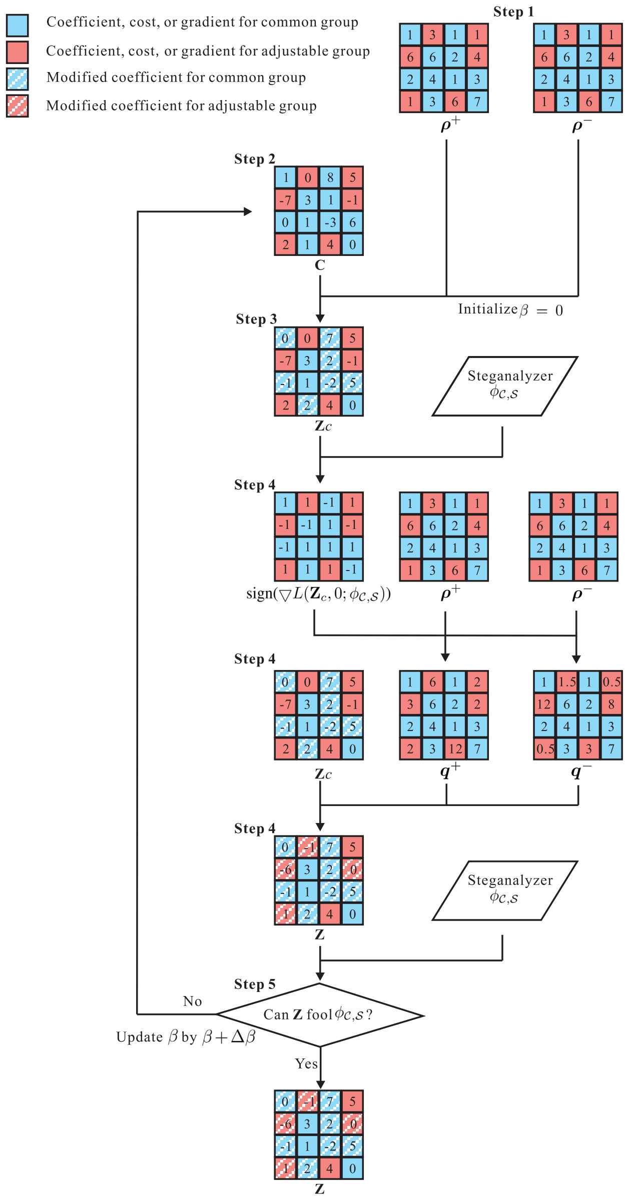


Fig. 1. Illustration of the process of the proposed ADV-EMB scheme.

composed of a fixed DCT filtering layer and 20 learnable convolutional layers. To the best of our knowledge, it achieves the best performance in detecting JPEG image steganography. In this paper, we use JPEG cover images and stego images generated by J-UNIWARD to train the target steganalyzer. However, other image formats, conventional embedding schemes or steganalyzers, may also be applicable, as indicated in Section III-A. The detailed steps of the proposed scheme are described as follows, and Fig. 1 illustrates an example.

1) For a cover image $\mathbf{C} = (c_{i,j})^{H \times W}$, use a conventional cost function (such as in J-UNIWARD) to compute the initial embedding costs, *i.e.*, $\{\rho^{+}_{i,j}, \rho^{-}_{i,j}\}$, for the DCT coefficients. Initialize the parameter $\beta = 0$.
2) Divide the elements in $\mathbf{C}$ into two disjoint groups, *i.e.*, a common group containing $l_1 = [H \times W \times (1 - \beta)]$ common elements, and an adjustable group containing $l_2 = H \times W - l_1$ adjustable elements. In Fig. 1, common group and adjustable group are labeled as blue and red boxes respectively. The positions of these two kinds of elements can be fixed in advance or randomized with the details to be discussed later.
3) Embed $k_1 = [k \times (1 - \beta)]$ bits into the common group using the initial embedding costs computed in Step 1 by applying a distortion minimization coding scheme, such as STC (syndrome-trellis codes) [45]. The resulting image is denoted as $\mathbf{Z}_c$. In Fig. 1, the modified coefficients in common group are highlighted with blue strides.
4) Compute the gradients $\nabla_{z_{i,j}} L(\mathbf{Z}_c, \hat{y}; \phi_{\mathcal{C},\mathcal{S}})$ of the steganalyzer using the target label $\hat{y} = 0$. Update the embedding costs for the adjustable elements by

$$q^{+}_{i,j} = \begin{cases} \rho^{+}_{i,j}/\alpha, & \text{if } -\nabla_{z_{i,j}} L(\mathbf{Z}_c, 0; \phi_{\mathcal{C},\mathcal{S}}) > 0, \\ \rho^{+}_{i,j}, & \text{if } -\nabla_{z_{i,j}} L(\mathbf{Z}_c, 0; \phi_{\mathcal{C},\mathcal{S}}) = 0, \\ \rho^{+}_{i,j}.\alpha, & \text{if } -\nabla_{z_{i,j}} L(\mathbf{Z}_c, 0; \phi_{\mathcal{C},\mathcal{S}}) < 0, \end{cases} \tag{12}$$

$$q^{-}_{i,j} = \begin{cases} \rho^{-}_{i,j}/\alpha, & \text{if } -\nabla_{z_{i,j}} L(\mathbf{Z}_c, 0; \phi_{\mathcal{C},\mathcal{S}}) < 0, \\ \rho^{-}_{i,j}, & \text{if } -\nabla_{z_{i,j}} L(\mathbf{Z}_c, 0; \phi_{\mathcal{C},\mathcal{S}}) = 0, \\ \rho^{-}_{i,j}.\alpha, & \text{if } -\nabla_{z_{i,j}} L(\mathbf{Z}_c, 0; \phi_{\mathcal{C},\mathcal{S}}) > 0, \end{cases} \tag{13}$$

where α is a scaling factor larger than 1 to ensure that equations(12) and (13) necessarily fulfill equation(10). α is set to 2 in this work. Embed $k_2 = k - k_1$ bits into the adjustable elements by using the updated embedding costs computed from (12) and (13) and the same coding scheme used for the common group. The resultant image is $\mathbf{Z}$. Figure 1 shows that the costs of the elements in the adjustable group are either doubled or halved, depending on the signs of the corresponding gradients. After data embedding, the modified coefficients in adjustable group are highlighted with red strides.
5) Take $\mathbf{Z}$ as the input of the steganalyzer $\phi_{\mathcal{C},\mathcal{S}}$. If $\phi_{\mathcal{C},\mathcal{S}}(\mathbf{Z}) = 0$, which means the adversarial stego $\mathbf{Z}$ can fool the steganalyzer with a minimum value of β, use $\mathbf{Z}$ as the output and terminate the embedding process. Otherwise, the amount of adjustable elements may not be enough. In this case, update β by $\beta + \Delta\beta$, and repeat Step 2 to Step 5 until $\beta = 1$. We use $\Delta\beta = 0.1$ in this work. If $\beta = 1$ and $\phi_{\mathcal{C},\mathcal{S}}(\mathbf{Z}) = 1$, which corresponds to the failure case of adversarial embedding, we just use a conventional steganographic scheme for embedding and output a conventional stego image.

Since the same coding scheme, such as STC, is used both in the adjustable group and the common group, the message receiver neither needs to be informed about the value of β, nor needs to know which image elements belong to to the adjustable group or the common group. Data is extracted in the same way as the baseline steganographic scheme.

As we know, in most existing steganographic schemes, an embedding order of image elements is generated by scrambling the indexes of image elements, where the scrambling operation is determined by a secret key shared between the sender and the receiver. The secret key can be fixed for different images, or changed as a session key. In the ADV-EMB implementation, the positions of the common elements and that of adjustable elements can be determined as follows. First, generate an embedding order in the same way as the baseline steganographic scheme. Then, the common group is formed by the first $l_1 = [H \times W \times (1-\beta)]$ elements according to the embedding order. Finally, the adjustable group is formed by the remaining elements. In other words, the positions of adjustable elements can be fixed or randomized for different images, depending on whether the embedding order is fixed or randomized. We recommend randomization for enhancing security.

IV. Experiments

In order to evaluate the performance of the proposed ADV-EMB scheme, we conducted the following experiments.

1) We evaluated the performance of ADV-EMB in the presence of an adversary-unaware steganalyst who trained his steganalyzer with conventional stego images. This corresponds to a white-box attack in adversarial examples [33] and it is the most favorable case for the steganographer. It will be reported in Section IV-B. We also evaluated the performance when non-target feature-based or CNN-based steganalyzers were used.
2) We evaluated the performance of ADV-EMB in the presence of an adversary-aware steganalyst who re-trained his steganalyzer with adversarial stego images. This corresponds to a challenging case for the steganographer. It will be reported in Section IV-C.
3) We simulated the situation when the knowledge of the steganographer and that of the steganalyst were alternatively updated. To the best of our knowledge, this is the first work to investigate iterative adversarial conditions for steganography and steganalysis. It will be demonstrated in Section IV-D.
4) Experimental results in Section IV-E will show why adversarial embedding guided by gradients and minimum amount of adjustable elements are important in the proposed scheme.
5) The role of randomizing the positions of the adjustable elements will be discussed in Section IV-G.
6) We performed some experiments on another image set for further evaluation, and the results will be shown in Section IV-H.
7) We evaluated the performance on spatial domain images, and the results will be given in Section IV-I.

The common settings and notations in the experiments will be described in Section IV-A. Some statistical information about the stego image sets will be provided in Section IV-F.

A. Settings

1) Image Set: The following two cover image sets were respectively used.

- Basic500k, denoted by $\mathcal{C}_B$. It was obtained by randomly selecting 5×10^5 JPEG images with size larger than 256×256 from ImageNet and then cropping their left top 256×256 regions. The images were further converted to grayscale and re-compressed into JPEG format with quality factor 75. This dataset has been used in [27] to train CNN-based steganalyzers. Although the images in Basic500K suffer from double/multiple JPEG compression, their use does not jeopardize the practical security of the embedded scheme. In fact, double/multiple JPEG compressed images are common in practice. Unless specified otherwise, the experiments were carried out on this image set. To use the images efficiently under different circumstances, $\mathcal{C}_B$ was randomly split into three disjoint subsets, $\mathcal{C}_B^0$, $\mathcal{C}_B^{1trn}$, and $\mathcal{C}_B^{1tst}$, with 2.5×10^5 images, 1.5×10^5, and 1×10^5 images, respectively. $\mathcal{C}_B^0$ was used to train the target steganalyzer, while $\mathcal{C}_B^{1trn}$ and $\mathcal{C}_B^{1tst}$ were used to generate adversarial stego images. Specifically, $\mathcal{C}_B^{1trn}$ and its stego counterparts were used for training adversary-aware steganalyzers, and $\mathcal{C}_B^{1tst}$ and its stego counterparts were used for testing the performance of both adversary-unaware and adversary-aware steganalyzers.
- JPEG-BOSSBase, denoted by $\mathcal{C}_J$. In order to verify the performance of ADV-EMB on an image set with distinct difference from $\mathcal{C}_B$, we generated this set without any possible double JPEG compression artifacts. We used *Photoshop CS6* for demosaicking the full-resolution raw images from the *BOSSBase* v1.01 image set [46] and then converted them into grayscale images. Later we downsampled the obtained images with a *Bicubic* kernel so that the smaller image dimension was 256. Then we central cropped the longer dimension and we got images of size 256×256. Finally, we compressed the images into JPEG format with quality factor 75 to obtain the JPEG-BOSSBase dataset. The experiments in Section IV-H were carried out on this image set. $\mathcal{C}_J$ was randomly split into two disjoint subsets, $\mathcal{C}_J^{trn}$ and $\mathcal{C}_J^{tst}$, each with 5000 images. Both $\mathcal{C}_J^{trn}$ and $\mathcal{C}_J^{tst}$ were used to generate adversarial stego images, and their roles are similar to $\mathcal{C}_B^{1trn}$ and $\mathcal{C}_B^{1tst}$, respectively.

2) Steganalyzers: Four different steganalyzers were used to evaluate the security of the steganographic schemes. The details are described as follows.

- Xu-CNN steganalyzer [26], denoted as ϕ. To the best of our knowledge, it is the best performing date-driven JPEG CNN steganalyzer. The 20-layer CNN steganalyzer was proposed by Xu, and we built the CNN structure and set all training parameters as in [26], with the only difference that the batch size was set to 100 during the training stage, with 50 cover images and their corresponding stego counterparts. The CNN model trained at the 100000-*th* iteration was used as the steganalyzer.
- Zeng-CNN steganalyzer [27], denoted as ϕ'. This deep learning steganalyzer involves two stages, a hand-crafted stage including quantization and truncation operation, and a learnable stage composed of three subset networks. We trained this steganalyzer with the same settings as in [27].
- GFR steganalyzer [20], denoted as ϕ''. It is based on 17000 histogram features generated with Gabor filters and an FLD ensemble classifier [21].

TABLE I

THE SECURITY PERFORMANCE (IN %) AGAINST AN ADVERSARY-UNAWARE STEGANALYZER

Steganalyzer	Steganography	Testing Set	0.1 bpnzAC			0.2 bpnzAC			0.3 bpnzAC			0.4 bpnzAC			0.5 bpnzAC		
			P_{fa}	P_{md}	P_e	P_{fa}	P_{md}	P_e	P_{fa}	P_{md}	P_e	P_{fa}	P_{md}	P_e	P_{fa}	P_{md}	P_e
$\phi_{\mathcal{C}_B^0,\mathcal{S}_B^0}$	J-UNIWARD [12]	$\{\mathcal{C}_B^{1tst},\mathcal{S}_B^{1tst}\}$	44.1	42.3	43.2	32.5	34.6	33.6	24.0	24.8	24.4	17.5	18.7	18.1	12.9	13.4	13.2
	ADV-EMB	$\{\mathcal{C}_B^{1tst},\mathcal{Z}_B^{1tst}\}$	44.1	92.5	**68.3**	32.5	98.6	**65.6**	24.0	99.3	**61.6**	17.5	99.6	**58.5**	12.9	99.5	**56.2**
$\phi'_{\mathcal{C}_B^0,\mathcal{S}_B^0}$	J-UNIWARD [12]	$\{\mathcal{C}_B^{1tst},\mathcal{S}_B^{1tst}\}$	46.4	46.5	46.4	38.1	39.6	38.8	32.6	32.3	32.4	27.4	23.1	25.2	20.8	18.7	19.7
	ADV-EMB	$\{\mathcal{C}_B^{1tst},\mathcal{Z}_B^{1tst}\}$	46.4	54.3	**50.3**	38.1	51.7	**44.9**	32.6	48.1	**40.3**	27.4	43.2	**35.3**	20.8	38.7	**29.7**
$\phi''_{\mathcal{C}_B^0,\mathcal{S}_B^0}$	J-UNIWARD [12]	$\{\mathcal{C}_B^{1tst},\mathcal{S}_B^{1tst}\}$	47.7	45.4	46.5	42.8	40.4	41.6	36.7	35.1	35.9	31.6	29.1	30.4	25.7	23.4	24.6
	ADV-EMB	$\{\mathcal{C}_B^{1tst},\mathcal{Z}_B^{1tst}\}$	47.7	47.1	**47.3**	42.8	45.1	**43.9**	36.7	43.2	**40.0**	31.6	38.9	**35.3**	25.7	36.2	**30.9**
$\phi'''_{\mathcal{C}_B^0,\mathcal{S}_B^0}$	J-UNIWARD [12]	$\{\mathcal{C}_B^{1tst},\mathcal{S}_B^{1tst}\}$	48.6	47.4	48.0	45.3	44.3	44.8	40.0	41.6	40.8	36.0	36.4	36.2	31.0	30.7	30.8
	ADV-EMB	$\{\mathcal{C}_B^{1tst},\mathcal{Z}_B^{1tst}\}$	48.6	47.9	**48.3**	45.3	45.9	**45.5**	40.0	44.8	**42.3**	36.0	40.2	**38.1**	31.0	35.3	**33.2**

- DCTR steganalyzer [19], denoted as ϕ'''. It is based on 8000 dimensional DCT residual features and an FLD ensemble classifier [21].

The steganalytic performance was evaluated by the missed detection rate as in (2), the false alarm rate as in (3), and the total error rate as in (4) .

3) Steganographic Schemes: We used two steganographic schemes to generate stego images.

- J-UNIWARD [12]: It was used as a baseline steganographic scheme. The embedding costs of DCT coefficients were calculated in the wavelet domain using a Daubechies wavelet filter bank. The corresponding stego image sets are denoted as $\mathcal{S}_B^0$, $\mathcal{S}_B^1$, $\mathcal{S}_J^{trn}$, and $\mathcal{S}_J^{tst}$.
- ADV-EMB: In the proposed scheme, J-UNIWARD was used to compute the initial embedding costs and perform the conventional embedding. The steganalyzer $\phi_{\mathcal{C}_B^0,\mathcal{S}_B^0}$ based on Xu-CNN was used as the target steganalyzer for adversarial embedding. The corresponding adversarial stego image sets are denoted as $\mathcal{Z}_B^0$, $\mathcal{Z}_B^1$, $\mathcal{Z}_J^{trn}$, and $\mathcal{Z}_J^{tst}$. The scaling parameter used in (12) and (13) was set to $\alpha = 2$, where we have tried $\alpha \in \{1.5, 2, 3, 5, 10\}$ and found only minor difference in performance.

The optimal embedding simulator [9] was employed for both J-UNIWARD and ADV-EMB. The Matlab implementation of J-UNIWARD was used[1]. Our proposed ADV-EMB scheme was implemented using TensorFlow with Python interface. The experiments were run on a NVIDIA Tesla K80 GPU platform. The embedding payload was measured by bits per non-zero cover AC DCT coefficient (bpnzAC) as in [12], [26], and [27]. In Section IV-B and IV-C, we conducted experiments on 0.1, 0.2, 0.3, 0.4, and 0.5 bpnzAC. For the rest of the experiments, we used 0.4 bpnzAC since the steganalyzers perform better on higher payloads.

B. Performance Against an Adversary-Unaware Steganalyst

In this part, we study the case where the knowledge of the steganalyzer is exposed to the steganographer, but the steganalyst is unaware of the adversarial operation and still uses the current steganalyzer. In particular, we assume that the Xu-CNN steganalyzer $\phi_{\mathcal{C}_B^0,\mathcal{S}_B^0}$, which has been trained on the image set $\{\mathcal{C}_B^0, \mathcal{S}_B^0\}$, is available to the steganographer. Note that the steganographer does not need to have $\{\mathcal{C}_B^0, \mathcal{S}_B^0\}$ given that the steganalyzer $\phi_{\mathcal{C}_B^0,\mathcal{S}_B^0}$ is known. The steganographer can use $\phi_{\mathcal{C}_B^0,\mathcal{S}_B^0}$ to generate an adversarial stego set $\mathcal{Z}_B^1$ from the cover set $\mathcal{C}_B^1$.

We would like to know how well does the steganalyzer $\phi_{\mathcal{C}_B^0,\mathcal{S}_B^0}$ perform on classifying $\{\mathcal{C}_B^{1tst}, \mathcal{Z}_B^{1tst}\}$ when compared to classifying $\{\mathcal{C}_B^{1tst}, \mathcal{S}_B^{1tst}\}$. The experimental results are reported in Table I and the better performed results are shown in bold. Note that under the same payload rate, the false alarm rate P_{fa} is the same for $\{\mathcal{C}_B^{1tst}, \mathcal{Z}_B^{1tst}\}$ and $\{\mathcal{C}_B^{1tst}, \mathcal{S}_B^{1tst}\}$, due to the fact that the steganalyzer was trained on $\{\mathcal{C}_B^0, \mathcal{S}_B^0\}$ but tested on $\mathcal{C}_B^{1tst}$, which is shared in $\{\mathcal{C}_B^{1tst}, \mathcal{Z}_B^{1tst}\}$ and $\{\mathcal{C}_B^{1tst}, \mathcal{S}_B^{1tst}\}$. However, we can observe that the missed detection rate P_{md} is much higher for $\mathcal{Z}_B^{1tst}$ than for $\mathcal{S}_B^{1tst}$. These results indicate that the adversarial stego images generated by ADV-EMB can effective evade detection by the target steganalyzer.

In order to investigate the case where the adversarial stego images are analyzed by steganalyzers other than the target one, we conducted experiments by using three advanced steganalyzers, *i.e.,* $\phi'_{\mathcal{C}_B^0,\mathcal{S}_B^0}$, $\phi''_{\mathcal{C}_B^0,\mathcal{S}_B^0}$, and $\phi'''_{\mathcal{C}_B^0,\mathcal{S}_B^0}$, to perform the same classification tasks. The experimental results reported in Table I show that the performance of these detectors on the adversarial stego images are, at least to some extent, worse than those obtained on the stego images generated by J-UNIWARD. Although being designed to fool a target steganalyzer, the ADV-EMB scheme shows a certain effectiveness also against non-target steganalyzers. We speculate that this adaptability to other steganalyzers is due to the transferability of adversarial stego images. As shown by recent studies [44], [47], adversarial examples can be *transferred* across different machine learning models as long as such models are used to carry out the same task. Our results indicate that this phenomenon also applies to steganalysis (at least in

[1] It is downloaded from http://dde.binghamton.edu/download/stego_algorithms/

TABLE II
THE SECURITY PERFORMANCE (IN %) AGAINST AN ADVERSARY-AWARE STEGANALYZER

Steganalyzer	**Steganography**	**Testing Set**	**0.1 bpnzAC**			**0.2 bpnzAC**			**0.3 bpnzAC**			**0.4 bpnzAC**			**0.5 bpnzAC**		
			P_{fa}	P_{md}	P_e	P_{fa}	P_{md}	P_e	P_{fa}	P_{md}	P_e	P_{fa}	P_{md}	P_e	P_{fa}	P_{md}	P_e
$\phi_{\mathcal{C}_B^{1trn},\mathcal{S}_B^{1trn}}$	J-UNIWARD [12]	$\{\mathcal{C}_B^{1tst},\mathcal{S}_B^{1tst}\}$	45.8	42.2	44.0	35.5	32.5	34.0	25.0	25.6	25.3	17.9	19.6	18.7	13.1	14.1	13.6
$\phi_{\mathcal{C}_B^{1trn},\mathcal{Z}_B^{1trn}}$	ADV-EMB	$\{\mathcal{C}_B^{1tst},\mathcal{Z}_B^{1tst}\}$	48.2	46.8	**47.5**	39.4	41.5	**40.4**	34.7	32.5	**33.6**	27.4	24.2	**25.8**	19.5	18.9	**19.2**
$\phi'_{\mathcal{C}_B^{1trn},\mathcal{S}_B^{1trn}}$	J-UNIWARD [12]	$\{\mathcal{C}_B^{1tst},\mathcal{S}_B^{1tst}\}$	45.7	47.8	46.7	37.5	41.3	39.4	34.2	31.5	32.9	23.3	27.2	25.1	20.1	19.6	19.8
$\phi'_{\mathcal{C}_B^{1trn},\mathcal{Z}_B^{1trn}}$	ADV-EMB	$\{\mathcal{C}_B^{1tst},\mathcal{Z}_B^{1tst}\}$	48.5	49.1	**48.8**	41.3	45.1	**43.2**	36.4	37.2	**36.8**	32.3	27.1	**29.7**	25.7	22.8	**24.2**
$\phi''_{\mathcal{C}_B^{1trn},\mathcal{S}_B^{1trn}}$	J-UNIWARD [12]	$\{\mathcal{C}_B^{1tst},\mathcal{S}_B^{1tst}\}$	48.4	45.0	46.7	42.8	40.5	41.7	37.1	35.1	36.1	31.3	29.5	30.4	25.3	24.1	24.7
$\phi''_{\mathcal{C}_B^{1trn},\mathcal{Z}_B^{1trn}}$	ADV-EMB	$\{\mathcal{C}_B^{1tst},\mathcal{Z}_B^{1tst}\}$	49.5	45.6	**47.4**	47.2	40.5	**43.7**	41.3	37.5	**39.4**	36.1	32.4	**34.2**	30.8	27.8	**29.3**
$\phi'''_{\mathcal{C}_B^{1trn},\mathcal{S}_B^{1trn}}$	J-UNIWARD [12]	$\{\mathcal{C}_B^{1tst},\mathcal{S}_B^{1tst}\}$	48.3	47.7	48.0	45.1	44.5	44.8	40.7	41.0	40.8	36.6	36.0	36.3	30.7	31.0	30.8
$\phi'''_{\mathcal{C}_B^{1trn},\mathcal{Z}_B^{1trn}}$	ADV-EMB	$\{\mathcal{C}_B^{1tst},\mathcal{Z}_B^{1tst}\}$	48.8	47.8	**48.3**	46.9	44.4	**45.7**	43.0	41.2	**42.1**	38.7	37.0	**37.9**	32.6	32.5	**32.6**

the framework studied in this paper). As a result, to counter an unknown steganalyzer, a steganographer may use a local well-performing CNN-steganalyzer as the target steganalyzer. Such an observation largely widens the use range of the proposed ADV-EMB scheme.

C. Performance Against an Adversary-Aware Steganalyst

In this part, we study the case where the steganalyst is aware of the adversarial embedding operation. As stated in Section II-D, one of his possible reactions is to re-train the steganalyzers with adversarial stego images. The adversarial stego images $\mathcal{Z}_B^{1trn}$ and $\mathcal{Z}_B^{1tst}$ were generated as in Section IV-B, where the steganographer only relies on the steganalyzer $\phi_{\mathcal{C}_B^0,\mathcal{S}_B^0}$ to generate adversarial stego images. Then, we trained the steganalyzers based on $\{\mathcal{C}_B^{1trn},\mathcal{Z}_B^{1trn}\}$ and tested on $\{\mathcal{C}_B^{1tst},\mathcal{Z}_B^{1tst}\}$. In this way, the image sets for data embedding (*i.e.,* $\mathcal{C}_B^{1trn}$ and $\mathcal{C}_B^{1tst}$) and that for the training target steganalyzer (*i.e.,* $\mathcal{C}_B^0$) were different, thus ensuring that ADV-EMB did not use any prior knowledge of the image set.

The experimental results we obtained are reported in Table II. It can be observed that compared to the target steganalyzer which is easily fooled by the adversarial stego images, a re-trained steganalyzer can better detect the adversarial embedding operations. However, compared to the baseline J-UNIWARD scheme, the proposed ADV-EMB scheme still achieves a better security performance. For example, ADV-EMB gets a 25.8% total error rate for 0.4 bpnzAC, which is comparable to J-UNIWARD with 25.3% for 0.3 bpnzAC. This means that under the same risk level of detection, ADV-EMB attains 0.1 bpnzAC more payload. As also shown in Table II, when we used the other three non-target steganalyzers ϕ', ϕ'', and ϕ''' for detection, higher total error rates are obtained on $\{\mathcal{C}_B^{1tst},\mathcal{Z}_B^{1tst}\}$ than on $\{\mathcal{C}_B^{1tst},\mathcal{S}_B^{1tst}\}$, showing once again that ADV-EMB outperforms the baseline scheme.

D. Sequential Iterative Process Between Steganographer and Steganalyst

In this part, we study a scenario wherein the steganographer and the steganalyst adjust their steganalyzer iteratively each time by adapting their knowledge about the scheme adopted by the adversary. It is assumed the steganographer takes the first step and then the steganalyst takes the second, together defining a round in the competing process.

The experiments were carried out under the following assumptions for each round. We assume that the steganographer uses a target steganalyzer which is trained by conventional stego images and adversarial stego images from all previous rounds, as it is reasonable to use all the available knowledge. Such a steganalyzer can be regarded as adversary-unaware for the current round (and also future rounds), but as adversary-aware for previous rounds. The adversary-aware steganalyst uses the latest generated adversarial stego images to train the steganalyzer. Such a steganalyzer may be more focused on detecting adversarial stego images generated in the current round.

We used the following experimental setting.

1) The current-round-adversary-unaware steganalyst is unaware of the adversarial stego images generated in the current round. For the first round, conventional stego images generated with the baseline steganographic scheme are used for training. For the subsequent rounds, the steganalyzer is trained on $\mathcal{C}_B^0$ and the corresponding adversarial stego images obtained in all previous rounds.
2) The steganographer sets the target steganalyzer to be the same as the adversary-unaware steganalyzer in the current round and tries to attack it by generating adversarial stego images from $\mathcal{C}_B^1$.
3) The current-round-adversary-aware steganalyst is aware of the adversarial operation performed in the current round. Hence the steganalyzer is trained on $\mathcal{C}_B^{1trn}$ and the adversarial stego counterpart in the current round.
4) To ease the comparison, the $\mathcal{C}_B^{1tst}$ and the corresponding adversarial stego counterpart are used to evaluate the performance for both the adversary-unaware steganalyzer and the adversary-aware steganalyzer. Each steganalyzer is used to detect stego images not only in the current round, but also in all previous rounds and future rounds.

TABLE III

THE STEGANALYTIC PERFORMANCE (IN %), GIVEN AS P_e (P_{fa}, P_{md}), IN THE ITERATIVE PROCESS WHEN THE STEGANOGRAPHER AND THE STEGANALYST ITERATIVELY UPDATE THEIR KNOWLEDGE OF THE OTHER SIDE

Round	**Steganalyzer**	**Testing set**			
		$\{\mathcal{C}_B^{1tst}, \mathcal{S}_B^{1tst}\}$	$\{\mathcal{C}_B^{1tst}, \mathcal{Z}_B^{1tst}\}$	$\{\mathcal{C}_B^{1tst}, \dot{\mathcal{Z}}_B^{1tst}\}$	$\{\mathcal{C}_B^{1tst}, \ddot{\mathcal{Z}}_B^{1tst}\}$
1	$\phi_{\mathcal{C}_B^0, \mathcal{S}_B^0}$	**17.94** (17.13, 18.75)	**58.26** (17.13, 99.39)	**34.28** (17.13, 51.43)	**33.36** (17.13, 49.58)
	$\phi_{\mathcal{C}_B^{1trn}, \mathcal{Z}_B^{1trn}}$	**21.29** (26.55, 16.03)	**25.60** (26.55, 24.65)	**30.81** (26.55, 35.06)	**28.92** (26.55, 31.29)
2	$\phi_{\mathcal{C}_B^0, \mathcal{Z}_B^0, \mathcal{S}_B^0}$	**20.51** (24.06, 16.95)	**25.40** (24.06, 26.73)	**60.72** (24.06, 97.37)	**28.57** (24.06, 33.07)
	$\phi_{\mathcal{C}_B^{1trn}, \dot{\mathcal{Z}}_B^{1trn}}$	**23.70** (30.62, 16.78)	**29.11** (30.62, 27.59)	**28.50** (30.62, 26.37)	**30.57** (30.62, 30.52)
3	$\phi_{\mathcal{C}_B^0, \dot{\mathcal{Z}}_B^0, \mathcal{Z}_B^0, \mathcal{S}_B^0}$	**24.82** (37.21, 12.43)	**27.81** (37.21, 18.40)	**29.41** (37.21, 21.60)	**66.22** (37.21, 95.23)
	$\phi_{\mathcal{C}_B^{1trn}, \ddot{\mathcal{Z}}_B^{1trn}}$	**22.22** (29.17, 15.27)	**26.92** (29.17, 24.66)	**30.57** (29.17, 31.96)	**27.75** (29.17, 26.32)

Although the iterative process can be endless, we performed three iteration rounds to illustrate the interplay between the steganographer and steganalyst. The current-round-adversary-unaware steganalyst used J-UNIWARD to generate the conventional stego image set S in the first round. The steganographer generated the adversarial stego sets Z, $\dot{Z}$, and $\ddot{Z}$ from the first to the third round, respectively. The embedding payload was set to 0.4 bpnzAC. The performances of current-round-adversary-unaware steganalyzer are shown in the first row of each round, and those of current-round-adversary-aware steganalyzer are shown in the second row of each round. From Table III, we can draw the following conclusions for the P_e.

1) Expectedly, the adversarial stego images generated in the current round can fool the current-round-adversary-unaware steganalyzer with the highest P_e. Compared to conventional stego images, all kinds of adversarial stego images achieve better security under the same steganalyzer. This implies that it is better to use adversarial stego images in any round.
2) For the current-round-adversary-unaware steganalyzers, as iterations go on, a steganalyzer in a higher round is less effective in detecting conventional stego images. Since the steganalyzers in higher rounds are trained not only on conventional stego images but also on adversarial stego images, the results may imply that the adversarial stego images in higher rounds disturb the current-round-adversary-unaware steganalyzer in detecting conventional stego images.
3) For the current-round-adversary-aware steganalyzer, although it is only trained with the adversarial stego images from the current round, it is also (more or less) effective to detect conventional stego images and adversarial stego images from other rounds. However, there is no clear trend to indicate whether it performs better on adversarial stego images from previous rounds or future rounds. For example, in Round 3, we can observe that the detection error rate is 27.75% for the current round, which is higher than 26.92% for the first round and lower than 30.57% for the second round. These results seem to show that adversarial stego images introduce somewhat similar modifications to fool the steganalyzer, no matter from which round.

E. Investigation on Two Important Steps in ADV-EMB

Performing adversarial embedding according to the inverse signs of gradients and using minimum amount of adjustable elements are the two most important steps of the ADV-EMB scheme. In this part, we investigate the effectiveness of each step. Both adversary-unaware and adversary-aware CNN steganalyzers were used for the evaluation, and the embedding payload was set to 0.4 bpnzAC.

1) Case I (Reversing the Signs in ADV-EMB): In the ADV-EMB scheme, the embedding costs of adjustable elements are asymmetrically adjusted according to the inverse signs of the gradients, as shown in (12) and (13). For comparison, we used the signs of the gradients, instead of the inverse signs, as in the following equations, to conduct experiments:

$$q_{i,j}^{+} = \begin{cases} \rho_{i,j}^{+}/\alpha, & \text{if } \nabla_{z_{i,j}} L(\mathbf{Z}_c, 0; \phi_{\mathcal{C},\mathcal{S}}) > 0, \\ \rho_{i,j}^{+}, & \text{if } \nabla_{z_{i,j}} L(\mathbf{Z}_c, 0; \phi_{\mathcal{C},\mathcal{S}}) = 0, \\ \rho_{i,j}^{+}.\alpha, & \text{if } \nabla_{z_{i,j}} L(\mathbf{Z}_c, 0; \phi_{\mathcal{C},\mathcal{S}}) < 0, \end{cases} \tag{14}$$

$$q_{i,j}^{-} = \begin{cases} \rho_{i,j}^{-}/\alpha, & \text{if } \nabla_{z_{i,j}} L(\mathbf{Z}_c, 0; \phi_{\mathcal{C},\mathcal{S}}) < 0, \\ \rho_{i,j}^{-}, & \text{if } \nabla_{z_{i,j}} L(\mathbf{Z}_c, 0; \phi_{\mathcal{C},\mathcal{S}}) = 0, \\ \rho_{i,j}^{-}.\alpha, & \text{if } \nabla_{z_{i,j}} L(\mathbf{Z}_c, 0; \phi_{\mathcal{C},\mathcal{S}}) > 0. \end{cases} \tag{15}$$

The results are shown in Table IV. Compared with the previous results (see Table I and II), the total error rate of the adversary-unaware steganalyzer drops from 58.5% to 18.3%, and that of the adversary-aware steganalyzer from 25.8% to 19.3%. The degraded performance indicates that taking into account the signs of the gradients plays an important role in producing the adversarial effect.

2) Case II (Disabling Minimum Amount of Adjustable Elements): In the ADV-EMB scheme, the number of adjustable elements is minimized through iteratively finding a minimum value of β for (11). In the comparative experiment, we used a fixed value of β for each image, and thus the amount of adjustable elements was the same for all the images. The results for $\beta = 0.1$, 0.3, and 0.5 are presented in Table IV. It can be observed that as β increases, the missed detection rate of the adversary-unaware steganalyzer increases, but the total error rate of the adversary-aware steganalyzer decreases. The results indicate that when increasing the number of adjustable elements, it becomes easier to fool the target steganalyzer. However, an excess of adversarial operations may

TABLE IV

THE SECURITY PERFORMANCE (IN %) WITH DIFFERENT SETTING FOR ADV-EMB UNDER THE PAYLOAD OF 0.4 BPNZAC

Steganalyzer	**Testing set**	**Case I**			**Case II ($\beta = 0.1$)**			**Case II ($\beta = 0.3$)**			**Case II ($\beta = 0.5$)**		
		P_{fa}	P_{md}	P_e	P_{fa}	P_{md}	P_e	P_{fa}	P_{md}	P_e	P_{fa}	P_{md}	P_e
$\phi_{\mathcal{C}_B^0,\mathcal{S}_B^0}$	$\{\mathcal{C}_B^{1tst},\mathcal{Z}_B^{1tst}\}$	17.5	19.2	18.4	17.5	49.2	33.3	17.5	87.0	52.3	17.5	96.2	56.9
$\phi_{\mathcal{C}_B^{1trn},\mathcal{Z}_B^{1trn}}$	$\{\mathcal{C}_B^{1tst},\mathcal{Z}_B^{1tst}\}$	18.2	20.5	19.3	23.9	22.8	23.3	23.0	22.7	22.8	18.1	19.4	18.7

TABLE V

THE FREQUENCIES OF OCCURRENCES OF β (IN %) IN GENERATING STEGO IMAGE SET $\mathcal{Z}_B^1$ FOR EACH PAYLOAD. THE SUM OF EACH COLUMN IS 100%

β	**0.1 bpnzAC**	**0.2 bpnzAC**	**0.3 bpnzAC**	**0.4 bpnzAC**	**0.5 bpnzAC**
0	40.61	34.08	24.74	18.60	13.35
0.1	13.31	22.00	28.15	31.67	32.71
0.2	9.83	16.65	22.56	26.21	28.08
0.3	7.85	11.00	12.70	13.47	14.73
0.4	6.13	6.50	5.98	5.68	6.39
0.5	4.70	3.79	2.71	2.33	2.56
0.6	3.62	2.16	1.27	0.95	1.02
0.7	2.78	1.26	0.66	0.40	0.40
0.8	2.09	0.76	0.35	0.22	0.19
0.9	1.50	0.43	0.20	0.10	0.09
1	0.06	0.02	0.01	0.01	0.01
fail	7.52	1.35	0.67	0.36	0.47

introduce unnecessary artifacts, leading to easier detection by an adversary-aware steganalyzer. Consequently, it is a better choice to use "just enough" amount of adjustable elements by balancing the performance of an adversary-unaware steganalyzer and an adversary-aware steganalyzer.

F. Supplementary Statistical Information

To further understand the proposed ADV-EMB scheme, we provide some supplementary statistical information on the adversarial stego images as follows.

1) Frequency of Adversarial Embedding Operation: To investigate the statistics on how many adjustable elements are used in the ADV-EMB scheme, the occurrences of β in generating the 2.5×10^5 adversarial stego images $\mathcal{Z}_B^1$ are given in Table V. Based on the statistics, we can make the following observations.

- For a low payload, such as 0.1 bpnzAC, since the steganalyzer is less effective in detecting conventional stego images, adversarial embedding is not necessary for a large portion of the stego images, which corresponds to the case of $\beta = 0$. As the payload increases, more stego images requires adversarial embedding ($\beta \neq 0$).
- A lower failure rate of adversarial embedding is obtained for a higher payload (from 7.52% on 0.1 bpnzAC to 0.47% on 0.5 bpnzAC). This is due to the fact that more elements are involved in modification as the payload increase. For instance, less than 2% elements are used for modification for 0.1 bpnzAC, while more than 11% elements are used for modification for 0.5 bpnzAC, as shown in Table VI. Note that the failure rate is exactly the same as $1 - P_{md}$ of the adversary-unaware CNN steganalyzer given in Table I.
- For all payloads, larger values of β occur less frequently than lower values. However, this phenomenon cannot be taken for granted since it may be due to the specific images, the baseline steganographic scheme, the target steganalyzer, and the step $\Delta\beta$ used to search the minimum β.

TABLE VI

THE MODIFICATION RATE COMPUTED AS THE CHANGE PER NON-ZERO AC DCT COEFFICIENT (IN %) FOR THE TWO STEGANOGRAPHIC SCHEMES UNDER DIFFERENT PAYLOADS

Steganography	**0.1 bpnzAC**	**0.2 bpnzAC**	**0.3 bpnzAC**	**0.4 bpnzAC**	**0.5 bpnzAC**
J-UNIWARD [12]	1.80	3.97	6.32	8.80	11.37
ADV-EMB	1.84	4.04	6.43	8.95	11.57

2) Modification Rate: In Section III-C, we have stated that adversarial embedding would lead to an increasing number of modified image elements due to the asymmetric costs assigned to the adjustable elements. We define the modification rate as the ratio of the number of changed coefficients to the total amount of non-zero AC DCT coefficients. In Table VI, we show the averaged modification rate for J-UNIWARD and ADV-EMB under different payloads on the image set $\mathcal{C}_B^1$. As expected, we can observe that the modification rates for ADV-EMB are slightly higher than for J-UNIWARD. Besides, the gap in the modification rate between J-UNIWARD and ADV-EMB widens as the payload increases (0.04%, 0.07%, 0.11%, 0.15%, 0.2% for the five payloads, respectively). This is due to the fact that more cases of $\beta \neq 0$ occur for a higher payload, as indicated in Table V. Please note that a higher modification rate may result in lower image quality, which may be a minor disadvantage of the proposed scheme.

G. Discussion on the Role of Randomizing the Positions of Adjustable Elements

In our previous experiments, the positions of adjustable elements are randomized by using different embedding orders for different images. One question is whether there is a difference in security performance between randomized positions and fixed positions. To answer the question, we conducted two comparative experiments and report the results in this part.

In the first experiment, we used a fixed embedding order for different images. As indicated in Section III-D, the fixed embedding order results in the fixed positions of adjustable elements. We adopted the same setting we have used in

TABLE VII

THE SECURITY PERFORMANCE (IN %), GIVEN IN P_e, OF ADV-EMB WITH A FIXED EMBEDDING ORDER AGAINST THE ADVERSARY-UNAWARE STEGANALYZER AND THE ADVERSARY-AWARE STEGANALYZER. THE TESTING IMAGE SET IS $\{\mathcal{C}_B^{1tst}, \mathcal{Z}_B^{1tst}\}$. PERFORMANCE COMPARISON WITH THE IMPLEMENTATION USING A RANDOMIZED EMBEDDING ORDER IS SHOWN IN THE PARENTHESIS

Steganalyzer	0.1 bpnzAC	0.2 bpnzAC	0.3 bpnzAC	0.4 bpnzAC	0.5 bpnzAC
$\phi_{\mathcal{C}_B^0,\mathcal{S}_B^0}$	68.32 (↓0.02)	65.61 (↓0.02)	61.68 (↑0.04)	58.58 (↑0.04)	56.24 (↑0.01)
$\phi_{\mathcal{C}_B^{1trn},\mathcal{Z}_B^{1trn}}$	47.49 (↓0.05)	40.60 (↑0.15)	33.05 (↓0.58)	25.18 (↓0.65)	19.05 (↓0.12)

TABLE VIII

THE SECURITY PERFORMANCE (IN %) OF ADV-EMB WITH A FIXED EMBEDDING ORDER AND A FIXED NUMBER OF ADJUSTABLE ELEMENTS ($\beta = 0.3$) AGAINST THE ADVERSARY-UNAWARE STEGANALYZER AND THE ADVERSARY-AWARE STEGANALYZER. THE TESTING IMAGE SET IS $\{\mathcal{C}_B^{1tst}, \mathcal{Z}_B^{1tst}\}$. PERFORMANCE COMPARISON WITH THE IMPLEMENTATION USING A RANDOMIZED EMBEDDING ORDER IS SHOWN IN THE PARENTHESIS

Steganalyzer	P_{fa}	P_{md}	P_e
$\phi_{\mathcal{C}_B^0,\mathcal{S}_B^0}$	17.5 (-)	87.2 (↑0.2)	52.4 (↓0.1)
$\phi_{\mathcal{C}_B^{1trn},\mathcal{Z}_B^{1trn}}$	14.1 (↓8.9)	15.1 (↓7.6)	14.6 (↓8.2)

TABLE IX

THE SECURITY PERFORMANCE (IN %) ON JPEG-BOSSBASE IMAGE SET UNDER THE PAYLOAD OF 0.4 BPNZAC

Steganalyzer	Steganography	Testing Set	P_{fa}	P_{md}	P_e
$\phi_{\mathcal{C}_J^0,\mathcal{S}_J^0}$	J-UNIWARD [12]	$\{\mathcal{C}_J^1, \mathcal{S}_J^1\}$	14.3	23.7	19.0
$\phi_{\mathcal{C}_J^0,\mathcal{Z}_J^0}$	ADV-EMB	$\{\mathcal{C}_J^1, \mathcal{Z}_J^1\}$	20.3	32.2	**26.3**
$\phi''_{\mathcal{C}_J^0,\mathcal{S}_J^0}$	J-UNIWARD [12]	$\{\mathcal{C}_J^1, \mathcal{S}_J^1\}$	20.7	21.4	21.1
$\phi''_{\mathcal{C}_J^0,\mathcal{Z}_J^0}$	ADV-EMB	$\{\mathcal{C}_J^1, \mathcal{Z}_J^1\}$	23.8	25.7	**24.7**
$\phi'''_{\mathcal{C}_J^0,\mathcal{S}_J^0}$	J-UNIWARD [12]	$\{\mathcal{C}_J^1, \mathcal{S}_J^1\}$	29.6	28.2	28.9
$\phi'''_{\mathcal{C}_J^0,\mathcal{Z}_J^0}$	ADV-EMB	$\{\mathcal{C}_J^1, \mathcal{Z}_J^1\}$	30.0	29.4	**29.7**

Section IV-B and IV-C. Adversary-unaware and adversary-aware CNN-based steganalyzers were respectively used for detection. The results we got are shown in Table VII. It can be observed that ADV-EMB with the fixed positions of adjustable elements and that with the randomized positions of adjustable elements do not have obvious difference in performance against the CNN-based steganalyzers. In the second experiment, we used a fixed embedding order and a fixed number of adjustable elements ($\beta = 0.3$) for each image. The payload was set to 0.4 bpnzAC. The results we got are given in Table VIII. It can be observed that the performance does not change much for an adversary-unaware steganalyzer, while it degrades greatly for an adversary-aware steganalyzer. This phenomenon is interesting. Although the fixed positions of adjustable elements are not directly leaked to the adversary-aware steganalyzer, the experimental evidence shows that the data-driven steganalyzer can automatically learn such information. In a similar scenario, when the same key is re-used for data embedding simulation, a CNN-based method [48] is highly effective in detecting different stego images with synchronized modification locations. The performance drops greatly when different keys are used for different images. The phenomenon does not occur for feature based steganalyzers. We speculate that modifications in the same location may present a chance of "collision attack" from the perspective of CNN-based steganalyzers. The neurons may learn strong activations from the synchronized modification positions. Since ADV-EMB employs minimum amount of adjustable elements, the collision effect is eliminated, even when a fixed embedding order is used, as the results reported in Table VII show.

H. Performance on JPEG-BOSSBase Image Set

In this part, we evaluate the performance of ADV-EMB on the image set JPEG-BOSSBase. The Xu-CNN steganalyzer $\phi_{\mathcal{C}_B^0,\mathcal{S}_B^0}$ trained on Basic500k was still used as the target steganalyzer in the ADV-EMB scheme and we generated adversarial stego images on JPEG-BOSSBase $\mathcal{C}_J$. We used three adversary-aware steganalyzers to detect ADV-EMB, and used J-UNIWARD as the baseline for comparison. The embedding payload was set to 0.4 bpnzAC. From the results shown in Table IX, we can observe that ADV-EMB performs better than J-UNIWARD on JPEG-BOSSBase. The results indicate that the good performance of the proposed ADV-EMB scheme does not rely much on a specific image set.

I. Experiments on Images in Spatial Domain

In this part, we investigate whether ADV-EMB can be extended to pixel domain staganography. The BOSSBase v1.01 image set [46], which contains 10000 grayscale 512 × 512 images, was used and denoted by $\mathcal{C}_{BS}$. We randomly split it into three disjoint subsets, $\mathcal{C}_{BS}^0$, $\mathcal{C}_{BS}^{1trn}$, and $\mathcal{C}_{BS}^{1tst}$, respectively with 5000, 2500, and 2500 images. The process of generating adversarial stego images is the same as in Section IV-A, except for the baseline steganographic scheme and the target steganalyzer. We selected S-UNIWARD [12] as the baseline steganographic scheme. The corresponding stego image sets are referred to as $\mathcal{S}_{BS}^0$, $\mathcal{S}_{BS}^{1trn}$, and $\mathcal{S}_{BS}^{1tst}$. Xu-Net [24], denoted as φ, was used as steganalyzer. This is a 6 layer CNN steganalyzer working in the spatial domain by using deep learning techniques, such as, batch normalization, 1×1 convolution, and global pooling. The Xu-Net steganalyzer trained on $\{\mathcal{C}_{BS}^0, \mathcal{S}_{BS}^0\}$ *i.e.,* $\varphi_{\mathcal{C}_{BS}^0,\mathcal{S}_{BS}^0}$ was used as the target steganalyzer. The corresponding adversarial stego images are denoted as $\mathcal{Z}_{BS}^0$, $\mathcal{Z}_{BS}^{1trn}$, and $\mathcal{Z}_{BS}^{1tst}$. The embedding payload is given in bits per pixel (bpp). A hand-crafted SRM (Spatial Rich Model) feature based steganalyzer equipped with an FLD ensemble classifier [16], denoted as φ', was used for performance evaluation. From Table X, we can observe that in the case of adversary-unaware steganalysis, the missed detection rate of ADV-EMB against target steganalyzer $\phi_{\mathcal{C}_{BS}^0,\mathcal{S}_{BS}^0}$ under

TABLE X

THE SECURITY PERFORMANCE (IN %) ON SPATIAL IMAGES AGAINST AN ADVERSARY-UNAWARE STEGANALYZER

Steganalyzer	Steganography	Testing Set	0.2 bpp			0.4 bpp		
			P_{fa}	P_{md}	P_e	P_{fa}	P_{md}	P_e
$\varphi_{\mathcal{C}_{BS}^0,\mathcal{S}_{BS}^0}$	S-UNIWARD [12]	$\{\mathcal{C}_{BS}^{1tst},\mathcal{S}_{BS}^{1tst}\}$	26.1	42.5	34.3	19.5	22.7	21.1
	ADV-EMB	$\{\mathcal{C}_{BS}^{1tst},\mathcal{Z}_{BS}^{1tst}\}$	26.1	98.4	**62.3**	19.5	100	**59.8**
$\varphi'_{\mathcal{C}_{BS}^0,\mathcal{S}_{BS}^0}$	S-UNIWARD [12]	$\{\mathcal{C}_{BS}^{1tst},\mathcal{S}_{BS}^{1tst}\}$	35.9	28.8	32.4	22.3	19.0	20.7
	ADV-EMB	$\{\mathcal{C}_{BS}^{1tst},\mathcal{Z}_{BS}^{1tst}\}$	35.9	33.3	**34.6**	22.3	33.7	**28.0**

TABLE XI

THE SECURITY PERFORMANCE (IN %) ON SPATIAL IMAGES AGAINST AN ADVERSARY-AWARE STEGANALYZER

Steganalyzer	Steganography	Testing Set	0.2 bpp			0.4 bpp		
			P_{fa}	P_{md}	P_e	P_{fa}	P_{md}	P_e
$\varphi_{\mathcal{C}_{BS}^{1trn},\mathcal{S}_{BS}^{1trn}}$	S-UNIWARD [12]	$\{\mathcal{C}_{BS}^{1tst},\mathcal{S}_{BS}^{1tst}\}$	34.3	39.6	37.0	21.4	24.1	22.8
$\varphi_{\mathcal{C}_{BS}^{1trn},\mathcal{Z}_{BS}^{1trn}}$	ADV-EMB	$\{\mathcal{C}_{BS}^{1tst},\mathcal{Z}_{BS}^{1tst}\}$	35.2	39.8	**37.5**	21.1	27.9	**24.5**
$\varphi'_{\mathcal{C}_{BS}^{1trn},\mathcal{S}_{BS}^{1trn}}$	S-UNIWARD [12]	$\{\mathcal{C}_{BS}^{1tst},\mathcal{S}_{BS}^{1tst}\}$	34.1	31.3	32.7	23.7	19.5	21.6
$\varphi'_{\mathcal{C}_{BS}^{1trn},\mathcal{Z}_{BS}^{1trn}}$	ADV-EMB	$\{\mathcal{C}_{BS}^{1tst},\mathcal{Z}_{BS}^{1tst}\}$	37.8	31.2	**34.5**	27.2	20.1	**23.7**

the payload 0.4 bpp reaches 100%, and that against $\phi'_{\mathcal{C}_{BS}^0,\mathcal{S}_{BS}^0}$ is 5%-12% higher than S-UNIWARD. From Table XI, we can observe that in the case of adversary-aware steganalyst, ADV-EMB outperforms S-UNWIARD by 2% on 0.4 bpp. The experimental results indicate that ADV-EMB can be easily extended to work in the pixel domain.

V. CONCLUSIONS

In this paper, we proposed a novel approach to look at the steganographic problem; namely, we proposed to embed the stego message while simultaneously taking into account the necessity of countering an advanced CNN-based steganalyzer. Such an aim is achieved by introducing a new adversarial embedding method, which takes both data embedding and adversarial operation into account. A practical steganographic scheme, ADV-EMB, which generates adversarial stego images with minimum amount of adjustable elements, has been illustrated to counter a deep learning based target steganalyzer. The extensive experiments we have carried out permitted us to reach the following conclusions:

1) When the target steganalyzer is accessible by the steganographer but the steganalyst is unaware of the adversary operation, a high missed detection rate can be achieved by ADV-EMB to counter the target steganalyzer.
2) When the steganalyst is aware of the adversarial embedding, and uses adversarial stego images to re-train the steganalyzer, the proposed ADV-EMB leads to a higher detection error rate compared to the state-of-the-art baseline steganographic scheme, for both target and non-target steganalyzers.
3) When both the steganographer and the steganalyst iteratively adjust their strategies according to the updated knowledge about the other side, adversarial stego images still have an advantage over their conventional counterparts.

Our approach to adversarial embedding shows a promising way to enhance steganographic security, still there are several unsolved issues to consider. To start with, the proposed ADV-EMB scheme uses only the signs of the gradients. It worths investigating whether the amplitudes of the gradients can also be helpful. Besides, it is worth studying on whether universal perturbations [49] are feasible in obtaining adversarial stego images. Furthermore, for a complete characterization of the interplay between the steganographer and the steganalyst, it would be interesting to resort to a game-theoretic formulation of the problem [38], [50], [51].

REFERENCES

[1] B. Li, J. He, J. Huang, and Y. Q. Shi, "A survey on image steganography and steganalysis," *J. Inf. Hiding Multimedia Signal Process.*, vol. 2, no. 2, pp. 142–172, Apr. 2011.

[2] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems: Breaking the steganographic utilities EzStego, Jsteg, Steganos, and S-Tools-and some lessons learned," in *Proc. Int. Workshop Inf. Hiding*, 1999, pp. 61–75.

[3] J. Fridrich and M. Goljan, "On estimation of secret message length in LSB steganography in spatial domain," in *Proc. SPIE*, vol. 5306, pp. 23–36, Jun. 2004.

[4] N. Provos, "Defending against statistical steganalysis," in *Proc. 10th Conf. USENIX Secur. Symp.*, vol. 10, 2001, pp. 323–336.

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

[6] W. Luo, F. Huang, and J. Huang, "Edge adaptive image steganography based on LSB matching revisited," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 201–214, Jun. 2010.

[7] T. Pevný, P. Bas, and J. Fridrich, “Steganalysis by subtractive pixel adjacency matrix,” *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 215–224, Jun. 2010.

[8] C. Chen and Y. Q. Shi, “JPEG image steganalysis utilizing both intrablock and interblock correlations,” in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 3029–3032.

[9] J. Fridrich and T. Filler, “Practical methods for minimizing embedding impact in steganography,” *Proc. SPIE*, vol. 6505, pp. 650502-1–650502-15, Jan. 2007.

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

[11] V. Holub and J. Fridrich, “Designing steganographic distortion using directional filters,” in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2012, pp. 234–239.

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

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

[14] L. Guo, J. Ni, and Y. Q. Shi, “Uniform embedding for efficient JPEG steganography,” *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 5, pp. 814–825, May 2014.

[15] W. Zhou, W. Zhang, and N. Yu, “A new rule for cost reassignment in adaptive steganography,” *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2654–2667, Nov. 2017.

[16] J. Fridrich and J. Kodovský, “Rich models for steganalysis of digital images,” *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 868–882, Jun. 2012.

[17] B. Li, Z. Li, S. Zhou, S. Tan, and X. Zhang, “New steganalytic features for spatial image steganography based on derivative filters and threshold LBP operator,” *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1242–1257, May 2018.

[18] J. Kodovský and J. Fridrich, “Steganalysis of JPEG images using rich models,” in *Proc. SPIE*, vol. 8303, p. 83030A, Feb. 2012.

[19] V. Holub and J. Fridrich, “Low-complexity features for JPEG steganalysis using undecimated DCT,” *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 219–228, Feb. 2015.

[20] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang, “Steganalysis of adaptive JPEG steganography using 2D Gabor filters,” in *Proc. 3rd ACM Workshop Inf. Hiding Multimedia Secur.*, 2015, pp. 15–23.

[21] J. Kodovský, J. Fridrich, and V. Holub, “Ensemble classifiers for steganalysis of digital media,” *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 432–444, Apr. 2012.

[22] S. Tan and B. Li, “Stacked convolutional auto-encoders for steganalysis of digital images,” in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA)*, Dec. 2014, pp. 1–4.

[23] Y. Qian, J. Dong, W. Wang, and T. Tan, “Deep learning for steganalysis via convolutional neural networks,” *Proc. SPIE*, vol. 9409, p. 94090J, Mar. 2015.

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

[25] G. Xu, H.-Z. Wu, and Y.-Q. Shi, “Ensemble of CNNs for steganalysis: An empirical study,” in *Proc. 4th ACM Workshop Inf. Hiding Multimedia Secur.*, 2016, pp. 103–107.

[26] G. Xu, “Deep convolutional neural network to detect J-UNIWARD,” in *Proc. 5th ACM Workshop Inf. Hiding Multimedia Secur.*, 2017, pp. 67–73.

[27] J. Zeng, S. Tan, B. Li, and J. Huang, “Large-scale JPEG image steganalysis using hybrid deep-learning framework,” *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1200–1214, May 2017.

[28] T. Denemark, P. Bas, and J. Fridrich, “Natural steganography in JPEG compressed images,” in *Proc. Electron. Imag.*, Jan. 2018, pp. 1–10.

[29] A. Nguyen, J. Yosinski, and J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 427–436.

[30] J. Bruna *et al.*, “Intriguing properties of neural networks,” in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–10.

[31] T.-T. Do, E. Kijak, L. Amsaleg, and T. Furon, “Enlarging hacker’s toolbox: Deluding image recognition by attacking keypoint orientations,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2012, pp. 1817–1820.

[32] Z. Chen, B. Tondi, X. Li, R. Ni, Y. Zhao, and M. Barni, “A gradient-based pixel-domain attack against SVM detection of global image manipulations,” in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Dec. 2017, pp. 1–6.

[33] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–11.

[34] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “DeepFool: A simple and accurate method to fool deep neural networks,” in *Proc. IEEE Conf Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2574–2582.

[35] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. (2017). “Towards deep learning models resistant to adversarial attacks.” [Online]. Available: https://arxiv.org/abs/1706.06083

[36] G. F. Elsayed *et al.* (2018). “Adversarial examples that fool both computer vision and time-limited humans.” [Online]. Available: https://arxiv.org/abs/1802.08195

[37] A. Kurakin, I. Goodfellow, and S. Bengio. (2016). “Adversarial examples in the physical world.” [Online]. Available: https://arxiv.org/abs/1607.02533

[38] M. Barni and F. Pérez-González, “Coping with the enemy: Advances in adversary-aware signal processing,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8682–8686.

[39] S. Kouider, M. Chaumont, and W. Puech, “Adaptive steganography by oracle (ASO),” in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2013, pp. 1–6.

[40] S. Kouider, M. Chaumont, and W. Puech, “Technical points about adaptive steganography by oracle (ASO),” in *Proc. 20th Eur. Signal Process. Conf.*, Apr. 2012, pp. 1703–1707.

[41] Y. Zhang, W. Zhang, K. Chen, J. Liu, Y. Liu, and N. Yu, “Adversarial examples against deep neural network based steganalysis,” in *Proc. 6th ACM Workshop Inf. Hiding Multimedia Secur.*, 2018, pp. 67–72.

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

[43] T. Denemark and J. Fridrich, “Improving steganographic security by synchronizing the selection channel,” in *Proc. 3rd ACM Workshop Inf. Hiding Multimedia Secur.*, 2015, pp. 5–14.

[44] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in *Proc. IEEE Eur. Symp. Secur. Privacy*, Mar. 2016, pp. 372–387.

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

[46] P. Bas, T. Filler, and T. Pevný, “‘Break our steganographic system’: The ins and outs of organizing BOSS,” in *Proc. Int. Workshop Inf. Hiding*, 2011, pp. 59–70.

[47] N. Papernot, P. McDaniel, and I. Goodfellow. (2016). “Transferability in machine learning: from phenomena to black-box attacks using adversarial samples.” [Online]. Available: https://arxiv.org/abs/1605.07277

[48] L. Pibre, J. Pasquet, D. Ienco, and M. Chaumont, “Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover sourcemismatch,” in *Proc. Media Watermarking, Secur., Forensics*, 2016, pp. 1–11.

[49] J. H. Metzen, M. C. Kumar, T. Brox, and V. Fischer. (2017). “Universal adversarial perturbations against semantic image segmentation.” [Online]. Available: https://arxiv.org/abs/1704.05712

[50] M. Barni and B. Tondi, “The source identification game: An information-theoretic perspective,” *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 3, pp. 450–463, Mar. 2013.

[51] M. Barni and B. Tondi, “Binary hypothesis testing game with training data,” *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4848–4866, Aug. 2014.

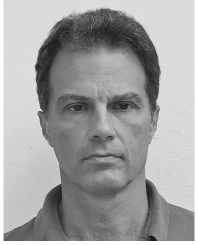
Weixuan Tang received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2014, where he is currently pursuing the Ph.D. degree with the School of Electronics and Information Technology. His research interests include digital image steganography, steganalysis, and deep learning.

Bin Li (S'07–M'09–SM'17) received the B.E. degree in communication engineering and the Ph.D. degree in communication and information system from Sun Yat-sen University, Guangzhou, China, in 2004 and 2009, respectively.

He was a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA, from 2007 to 2008. In 2009, he joined Shenzhen University, Shenzhen, China, where he is currently an Associate Professor. He is also the Director of the Shenzhen Key Laboratory of Media Security. His current research interests include image processing, multimedia forensics, and pattern recognition. He is a member of the IEEE Information Forensic and Security Technical Committee.

Shunquan Tan (M'10–SM'17) received the B.S. degree in computational mathematics and applied software and the Ph.D. degree in computer software and theory from Sun Yat-sen University, Guangzhou, China, in 2002 and 2007, respectively.

He was a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA, from 2005 to 2006. He is currently an Associate Professor with the College of Computer Science and Software Engineering, Shenzhen University, China. He is also a member of the Guangdong Key Lab of Intelligent Information Processing and Shenzhen Key Laboratory of Media Security. His current research interests include steganography, steganalysis, multimedia forensics, and deep machine learning.



Mauro Barni (MâŁ™92–SMâŁ™06–FâŁ™12) graduated in electronic engineering from the University of Florence in 1991. He received the Ph.D. degree in informatics and telecommunications in 1995. During the last two decades, he has been studying the application of image processing techniques to copyright protection and authentication of multimedia, and the possibility of processing signals that have been previously encrypted without decrypting them. He is currently a Professor with the Department of Information Engineering and Mathematics, University of Siena, Italy. He has been focusing on theoretical and practical aspects of adversarial signal processing. He has authored or co-authored about 300 papers published in international journals and conference proceedings, and holds five patents in the field of digital watermarking and image authentication. He has co-authored the book *Watermarking Systems Engineering: Enabling Digital Assets Security and other Applications* (Dekker Inc., 2004). He is a member of EURASIP. He was a recipient of the Individual Technical Achievement Award of EURASIP for 2016. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY from 2015 to 2017. He was the Funding Editor of the *EURASIP Journal on Information Security*. He served as an associate editor for many journals including several IEEE TRANSACTIONS. He was the Chairman of the IEEE Information Forensic and Security Technical Committee from 2010 to 2011. He was the Technical Program Chair of ICASSP 2014. He was appointed DL of the IEEE SPS from 2013 to 2014.

Jiwu Huang (M'98–SM'00–F'16) received the B.S. degree from Xidian University, Xi'an, China, in 1982, the M.S. degree from Tsinghua University, Beijing, China, in 1987, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Science, Beijing, in 1998. Since 2000, he has been with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China. He is currently a Professor with the College of Information Engineering, Shenzhen University, Shenzhen, China. His current research interests include multimedia forensics and security. He serves as a member for the IEEE CASS Multimedia Systems and Applications Technical Committee and the IEEE SPS Information Forensics and Security Technical Committee. He is an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.