Erase and Repair: An Efficient Box-Free Removal Attack on High-Capacity Deep Hiding

Hangcheng Liu, Tao Xiang, *Senior Member, IEEE*, Shangwei Guo, *Member, IEEE*, Han Li, Tianwei Zhang, *Member, IEEE*, and Xiaofeng Liao, *Fellow, IEEE*

***Abstract*— Deep hiding, embedding images with others using deep neural networks, has demonstrated impressive efficacy in increasing the message capacity and robustness of secret sharing. In this paper, we challenge the robustness of existing deep hiding schemes by preventing the recovery of secret images, building on our in-depth study of state-of-the-art deep hiding schemes and their vulnerabilities. Leveraging our analysis, we first propose a simple box-free removal attack on deep hiding that does not require any prior knowledge of the deep hiding schemes. To improve the removal performance on the deep hiding schemes that may be enhanced by adversarial training, we further design a more powerful removal attack, efficient box-free removal attack (EBRA), which employs image inpainting techniques to remove secret images from container images. In addition, to ensure the effectiveness of our attack and preserve the fidelity of the processed container images, we design an erasing phase based on the locality of deep hiding to remove secret information and then make full use of the visual information of container images to repair the erased visual content. Extensive evaluations show our method can completely remove secret images from container images with negligible impact on the quality of container images.**

***Index Terms*— Removal attack, deep hiding, high-capacity, image inpainting, adversarial training.**

I. Introduction

DATA hiding [1] is the art of concealing secret data within a cover image or other multimedia signals imperceptibly. It has gained popularity in applications such as secret communication [2], copy-right protection [3], and content authentication [4]. Typically, a data hiding scheme comprises two necessary algorithms: *hiding* and *revealing*. The hiding algorithm is responsible for embedding a secret message within a cover image without affecting its visual perception, transforming the cover image becomes a container image after hiding. The revealing algorithm recovers the embedded secret message from the container image.

Traditional data hiding schemes [5], [6], [7] have mainly focused on concealing binary messages and pursuing perfect revealing (i.e. revealed secret message is the same as the hidden one). However, one problem with these traditional methods is their very limited message capacity. For instance, a well-known data hiding scheme HUGO [5] only achieves less than 0.5 bits per pixel (bpp). Such a low message capacity hinders the effectiveness and application of data hiding, particularly when one needs to share massive secret information (e.g. images) through public channels.

Recently, researchers have applied deep neural networks (DNNs) to data hiding [8], [9], [10] due to the impressive performance of deep learning in various fields [11], [12], [13]. DNN-based deep hiding (i.e. deep hiding) benefits from the representational capacity of DNNs and increases the embedding rate to a surprising level (more than 24 bpp). Specifically, in [14], [15], and [16], the authors employ DNNs to perform the hiding and revealing algorithms and successfully hide a full-size image within another one (i.e. 24 bpp). UDH [15] even embeds up to three full-size images within one image (i.e. 72 bpp) using three pairs of DNNs. Such a significant increase in embedding rate makes deep hiding more practical and greatly improves the efficiency of secret sharing. In this paper, we focus on high-embedding-rate deep hiding schemes, where both the secret and cover are images, and study their robustness.

Existing study claims that existing deep hiding schemes not only have extremely high message capacity but also possess strong robustness, especially the data hiding models that are enhanced by adversarial training [15], [4], [17]. Then, an important question arises: *is this robustness sufficient to deal with potential disturbances?* Unfortunately, to the best of our knowledge, there are few studies proposed specifically to threaten the robustness of deep hiding. Jung et al. [18] proposed a removal attack that erases the embedded secret images from the corresponding container image by analyzing the pixel distributions. However, their attack only considers basic deep hiding models, which are not enhanced by adversarial training and fail to attack robust-enhanced deep hiding schemes. Besides, their method is inefficient for images

Manuscript received 4 April 2023; revised 27 July 2023 and 4 August 2023; accepted 9 August 2023. Date of publication 15 August 2023; date of current version 29 August 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3103500; in part by the National Natural Science Foundation of China under Grant U20A20176 and Grant 62072062; in part by the Natural Science Foundation of Chongqing, China, under Grant cstc2022ycjh-bgzxm0031 and Grant cstc2021jcyj-msxmX0273; in part by the Sichuan Science and Technology Program under Grant 02160024420003; in part by the Fundamental Research Funds for the Central Universities under Grant 2022CDJHLW005; and in part by the Singapore Ministry of Education (MOE) under Grant AcRF Tier 2 MOE-T2EP20121-0006 and Grant AcRF Tier 1 RS02/19. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Patrick Bas. *(Corresponding author: Tao Xiang.)*

Hangcheng Liu and Tianwei Zhang are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: hangcheng.liu@ntu.edu.sg; tianwei.zhang@ntu.edu.sg).

Tao Xiang, Shangwei Guo, Han Li, and Xiaofeng Liao are with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: txiang@cqu.edu.cn; swguo@cqu.edu.cn; lihan@cqu.edu.cn; xfliao@cqu.edu.cn).

Digital Object Identifier 10.1109/TIFS.2023.3305839

with large resolutions, as they complete the removal pixel by pixel.

In this paper, we challenge the robustness of existing deep hiding schemes in a *box-free* setting (the specific deep hiding method is unknown to the attacker, and all involved DNNs cannot be accessed), especially the schemes whose robustness is enhanced by adversarial training. In the remaining content, we aim to answer the following two questions: 1) *how robust are the existing deep hiding schemes?*, and 2) *can the secret information embedded by deep hiding be completely erased?*

To address the first question, we conduct a series of explorations and identify two vulnerabilities of existing deep hiding schemes: *locality* and *low redundancy*. Inspired by these vulnerabilities, we develop a simple and box-free lattice attack to answer the second question. Experimental results show that the lattice attack successfully prevents the recovery of secret images of basic hiding models. Unfortunately, we also observe that the lattice attack slightly reduces the visual quality of container images and fails to attack robust hiding models.

To improve the removal ability and preserve the visual quality of container images, we design a novel removal attack, named **E**fficient **B**ox-free **R**emoval **A**ttack (EBRA), to delete secret images from container images. EBRA mainly consists of two phases: erasing and repair. In the erasing phase, pixels in selected small regions of the container image are erased by setting them to 0. Based on the vulnerabilities we observed, the erasing operation completely removes the corresponding subregions of the secret image from the container image (see Section IV-A). To complete all missing regions, we design a quality-enhanced inpainting algorithm, which extracts the contour and color maps from the complete container image to assist an inpainting model to repair the incomplete image through feature fusion. Specifically, we propose two auxiliary models to extract the auxiliary maps. Once all subregions of the container image are processed, we combine all repaired regions to generate the final purified container image that is close to the original one and contains no secrets.

We conduct extensive experiments to evaluate the effectiveness of our EBRA. To accurately evaluate the removal effect, we conduct objective and subjective evaluations. In the objective evaluation, we use the existing image quality metrics to quantify the attack performance. For the subjective evaluation, we recruit 50 participants and asked them to judge whether the secret images are completely removed through visual observation. Both the objective and subjective evaluations confirm that our EBRA can remove embedded secrets blindly as well as preserve the visual quality of containers, even for robust-enhanced deep hiding models.

In summary, the contributions of this paper are:

- We observe two vulnerabilities (locality and low redundancy) of existing high-capacity deep hiding schemes with different meta-architectures.
- We take advantage of observed vulnerabilities and propose an effective removal attack that can remove embedded secret images efficiently and blindly.
- We conduct extensive experiments, including objective and subjective evaluations, to demonstrate the effectiveness of our EBRA.

The rest of this paper is organized as follows. Section II provides a summary of existing hiding methods. Section III formalizes deep hiding, removal attack, and the threat model. Section IV introduces general vulnerabilities of existing deep hiding methods and our first attempt to remove the embedded secret images. Section V describes our proposed removal attack. Section VI presents all experimental results and the corresponding analysis. Section VIII concludes this paper.

II. Related Work

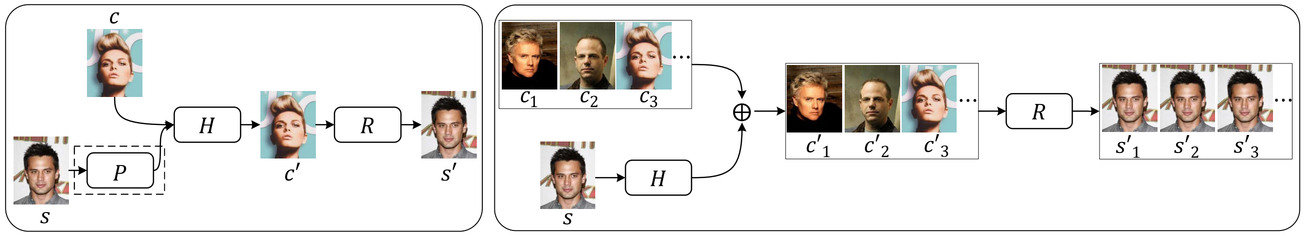
A. Data Hiding

In a complete secret transmission, the sender uses a hiding algorithm to embed a secret message within a cover medium (the cover medium is an image in this paper) that appears harmless. Only those who know the revealing algorithm and corresponding key can extract the secret message from the container image. A qualified data hiding scheme needs to meet two primary goals: (1) the hiding cannot affect the visual quality of the cover images, and (2) the revealed message should be consistent with the embedded one.

1) Deep Hiding: Deep hiding allows researchers to obtain the hiding and revealing algorithms in an end-to-end way and realize a much higher embedding rate [1]. Baluja [14] first proposed a framework to hide a full-size image within another one using DNNs. Specifically, the secret image is first processed by a prep network, and then the preprocessing result and cover image are fed into a hiding network that outputs the corresponding container image. In the revealing phase, the embedded secret image is extracted from the container image by a revealing network. This framework is called *cover-dependent deep hiding* (DDH) since its encoding process of secret images is cover dependent, shown as Fig. 1(a). Based on this framework of DDH, Zhang et al. [16] further proposed to hide a secret image in the Y channel of the cover image for improving the invisibility of the hiding. Yu [19] also tried to improve the quality of generated container images by adopting an attention model. At the same time, researchers also study how to further increase the message capacity using invertible neural networks under DDH framework. Specifically, methods proposed in [20] and [21] realized multiple images hiding by increasing the number of channels of the hidden branch.

Besides DDH, there is another meta-architecture for deep hiding that is termed *universal deep hiding* (UDH) [15] (see Fig. 1(b)). In UDH, the encoding process is cover independent, and the encoded secret can be arbitrarily hidden in different cover images. What's more, Zhang et al. [15] confirmed that the coding results of different hiding networks in UDH can be superimposed and embedded in one cover image, which also increases the message capacity.

2) Robustness: The revealing process of deep hiding can be easily destroyed by common image distortions. To address this issue, adversarial training [15], [4], [17] is often adopted to enhance the robustness, which uses noise layers to distort the original container images and requires the revealing model to recover correct secret images from distorted results. By now, adversarial training is the most effective way to avoid the influence of known distortions. Based on adversarial



(a) DDH (b) UDH

Fig. 1. Meta-architectures of existing deep hiding. (a) the hiding network (H) in DDH is responsible for encoding the secret image s and embedding the encoding results into the cover image c at the same time; (b) the hiding network in UDH is only responsible for encoding the secret image and the encoding results can be hidden in arbitrary cover images by subsequent element-wise addition.

training, Xu et al. [22] further proposed a content-aware noise projection (CANP) module that implicitly preserves essential information for the revealing. Luo et al. [23] combined adversarial training and channel coding to improve the robustness against agnostic distortions. Zhang et al. [17] further explored the gain brought by adversarial training from the perspective of forward and backward propagation and concluded that the main influential component is forward propagation.

B. Steganalysis

In the early study, steganalysis plays an important role in defeating steganography by determining whether a given image carries potential confidential information. Compared with classical steganalysis methods [24], [25], recently proposed DNN-based steganalysis methods not only simplify the feature design but also significantly increase the detection accuracy. For example, Ye et al. [26] proposed to initialize CNN's first layer with filters used in a spatial rich model [27] and designed a new activation function called truncated linear unit. Boroumand et al. [28] got rid of heuristics and externally enforced elements previously proposed for steganalysis and trained a deep residual network in an end-to-end way from randomly initialized parameters. You et al. [29] assumed that natural image noise is similar between different image subregions and adopted a Siamese CNN to determine the relationships between the noise of different subregions.

Although DNN-based steganalysis methods have shown high detection accuracy, there is still a miss rate which may be unacceptable when the confidentiality of sensitive information is extremely important. Besides, many ideas have been proposed to bypass passive detection. For example, in [30] and [31], adversarial training is used for secure data hiding, in which a potential adversary (steganalyzer) is considered, and the hiding network is trained to deceive the adversary. Another approach to combat steganalysis involves combining adversarial examples with data hiding, which tries to mislead the target steganalyzer as demonstrated in [32].

C. Removal Attack

Different from passive steganalysis which threatens the security of data hiding, removal attack (also known as image sterilization [33] or active steganalysis [18]) is an active action, threatening the availability of data hiding, which aims to erase potential secrets from container images while keeping their visual quality. Existing related study mainly focuses on traditional data hiding. For example, a simple bit-flipping function can destroy most stego pixels [33] embedded by low-capacity LSB-based data hiding methods. Imon et al. [34] proposed to use the pixel eccentricity property of suspected cover images to remove embedded secrets, which can be applied to pixel-value differencing steganography [35] besides LBS-based steganography. Ganguly and Mukherjee [36] proposed to selectively corrupts the integer wavelet coefficients of a given image to destroy the information hidden within it. Corley et al. [37] proposed to train a purifier (i.e. a DNN) based on collected quantities of container images. However, this way is not practical in the real world since it is hard to collect the container images once the deep hiding scheme is unknown. Researchers also tried to design different filters to erase steganography noise [38], [39]. All the above methods are designed for removing short binary messages hidden by classical data hiding methods and are inapplicable for recently proposed high-capacity deep hiding schemes.

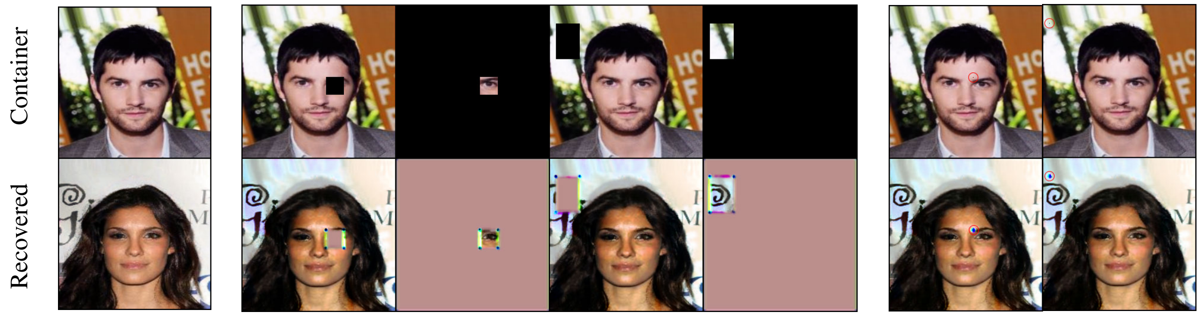
To the best of our knowledge, there are few works proposed specifically for removing secret images from container images in deep hiding. The work most relevant to our research is Pixel Steganalysis proposed in [18], which analyzes the distribution of each pixel in the image and adjusts suspicious pixels to get a purified container image. However, this approach is inefficient as secret information is removed pixel by pixel, and its effectiveness in dealing with robust deep hiding has not been verified. In this paper, we challenge the robustness of high-capacity deep hiding schemes under different meta-architectures in a blind way, especially the robust-enhanced deep hiding schemes.

III. Problem Formulation And Threat Model

A. Problem Formulation

We denote the hiding and revealing models in deep hiding as R and H. The hiding and revealing processes can be represented as $c' = H(c, s)$ and $s' = R(c')$, where c, s, c', and s' are the cover image, secret image, container image, and recovered secret image, respectively. As described in Section II, deep hiding has two goals, which are (1) $\mathbb{E}_{\sim s,c} D(c, c') \leq \xi_1$ and (2) $\mathbb{E}_{\sim s,c} D(s, s') \leq \xi_2$, where D measures the visual distance between two images, and ξ_1 and ξ_2 represent acceptable thresholds.

Our removal attack against deep hiding also has two goals: (1) to preserve the visual quality of container images, and



(a) Original (b) Locality (c) Low redundancy

Fig. 2. Vulnerabilities in existing deep hiding methods. (a) the original container and its corresponding recovered secret image. (b) we select random subregions in the original container and remove or only keep them to get new container images. (c) we reduce the removed subregion in each container to one pixel (mark with a red circle).

(2) to ensure no valid secret information can be observed after attacks (i.e., the visual distance should be large enough between the recovered secret images before and after the removal attack). Based on the two goals, we formally define the purified container images ($\hat{c'}$) liking

$$\max_{\hat{c'}} D(R(c'), R(\hat{c'}))\ s.t.\ D(c', \hat{c'}) \leq \xi_3, \tag{1}$$

where ξ_3 is also a threshold.

B. Threat Model

We assume that the specific deep hiding scheme is unknown and, so the accesses to H and R are impossible. This is a very strict assumption that ensures the practicality of our removal attack in the real world. In the field of adversarial attack [40], [41], there are two relatively loose assumptions: the target model can be accessed in a black-box way or in a white-box way, corresponding to black-box attack and white-box attack. Therefore, we call a removal attack that meets our strict assumption *box-free attack*.

IV. Observations and First Attempt

A. Vulnerabilities of Deep Hiding

Despite the variety of meta-architectures adopted in deep hiding, current deep hiding schemes have similar vulnerabilities regarding robustness. This is because most of the existing deep hiding schemes rely on fully convolutional networks, and hide significant amounts of information. Due to the local nature of the convolution operation, the influence of one secret pixel is limited to only its surrounding pixels, i.e., locality. Moreover, hiding significant amounts of information means that it is difficult to achieve redundancy, i.e., low redundancy. In the following content, we will elaborate on these vulnerabilities and provide visualization examples.

1) Locality: For each pixel s_{ij} of s, current deep hiding schemes trend to embed it in a tiny region (the size of the region is determined by receptive field) of the corresponding pixel c'_{ij} in c'. We illustrate the locality vulnerability of a DDH scheme DS [14] in Fig. 2(b). This vulnerability also exists in other DDH and UDH schemes, even the model's robustness is enhanced by adversarial training. From Fig. 2(b), we can observe that once a region of c' is removed, it becomes impossible to recover the corresponding secret pixels.

2) Low Redundancy: As we reduce the number of removal pixels in c' to one, we observe that a tiny region of the secret image cannot be recovered correctly (see Fig. 2(c)). We believe that this phenomenon is caused by high message capacity. The locality indicates that the information of each secret pixel is stored in a tiny region surrounding the corresponding container pixel. So each container pixel needs to carry the information of multiple secret pixels. However, the payload of each container pixel is limited due to the requirement of $\mathbb{E}_{\sim s,c} D(c, c') \leq \xi_1$. Hence, it is hard to ensure that the information stored on each container pixel is redundant, especially facing such a large amount of secret information as full-size images. Thus, the loss of one container pixel affects the recovery of multiple secret pixels around it, i.e., the marked blue areas in Fig. 2(c).

B. Lattice Attack

Inspired by the vulnerability of low redundancy, we first design a simple lattice attack that only distorts part pixels of c'. Specifically, we choose one pixel of c' every q pixels and change all three channels of chosen pixels as $c'_{mn} = (r_1, r_2, r_3)$ where m and n modulo $q+1$ equals 0, r_1, r_2, and r_3 are new random values for different channels. By setting an appropriate value of q, no valid secret information can be revealed due to the vulnerability of low redundancy. In experiments, we set q as 5 and test the lattice attack on UDH [15], DS [14], and ISGAN [16].

1) Metric: To evaluate the effectiveness of removal attacks, we use PSNR and VIF [42] as the distance metrics. *The higher the scores of both PSNR and VIF, the smaller the distance between images.* It should be noted that PSNR works well for relatively high-quality images but may not perform well on images with low visual quality [43], [44]. And VIF is closer to the human visual system on low-quality images than other mainstream metrics (e.g. PSNR and SSIM) [42], [45]. *Given that $R(\hat{c'})$ is often low-quality (see Fig. 3 and 7), VIF is more accurate than PSNR for evaluating the performance on preventing the revealing of secrets in this case.* In the following sections, we use PSNR-C, VIF-C, PSNR-S, and VIF-S to denote PSNR(c',$\hat{c'}$), VIF(c',$\hat{c'}$), PSNR($R(c')$,$R(\hat{c'})$), and VIF($R(c')$,$R(\hat{c'})$), respectively.

We present the average results of PSNR-C, VIF-C, PSNR-S, and VIF-S in Table I and provide some visualization samples

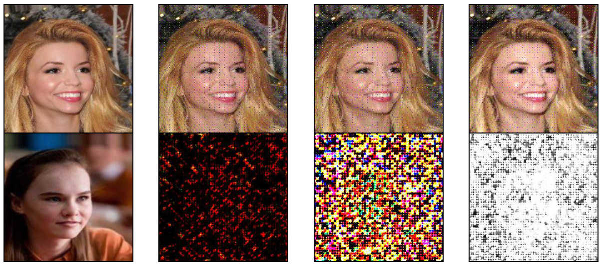


Fig. 3. Visualization examples of applying the lattice attack on three deep hiding methods. The container images (top row) are processed using the lattice attack, resulting in no valid secret information in the corresponding recovered secret images (bottom row).

Fig. 4. Lattice attack on Gaussian noise-based enhanced UDH. The images of the top row are the original secret images. The images of the bottom row are the corresponding recovered secret images after attacking.

TABLE I
EVALUATION RESULTS OF LATTICE ATTACK

Metric	UDH	DS	ISGAN
PSNR-C|PSNR-S	19.91|6.44	19.06|5.41	19.08|4.98
VIF-C|VIF-S	0.310|0.024	0.300|0.012	0.306|0.014

TABLE II
EVALUATION OF THE LATTICE ATTACK ON GAUSSIAN NOISE-BASED ENHANCED UDH

PSNR-C	PSNR-S	VIF-C	VIF-S
19.91	29.94	0.310	0.247

in Fig. 3. As expected, the lattice attack does prevent the revealing of embedded secret images. However, it fails to balance the two goals of removal attacks since the PSNR-C scores below 20 (Table I). One way to increase the image quality is to increase q. But it is hard to determine an appropriate q when H and R cannot be accessed. Besides the problem of image quality, the lattice attack cannot attack robust-enhanced models, especially those that are enhanced by Gaussian noise. We confirm this phenomenon in Table II and Fig. 4. From Fig. 4, we can see that even after applying the lattice attack to attack the enhanced UDH, a large amount of valid secret information can be still recovered.

V. METHODOLOGY

We have attempted to attack deep hiding only for its low redundancy, but we found that the simple lattice attack is ineffective when it faces more robust deep hiding models. Therefore, in this section, we propose a new attack, **E**fficient **B**ox-free **R**emoval **A**ttack (EBRA), to achieve the two attack goals described in Section III.

A. Framework Overview

1) Insight: Our EBRA uses image inpainting technology [46], [47], [48] and consists of two phases to remove embedded secret information from c' while preserving the quality of $\hat{c'}$. Our design strategies are two-fold. *First*, erasing pixels of c' must remove the corresponding secret pixels due to the locality vulnerability. *Second*, image inpainting can recover the missing regions for maintaining the quality of $\hat{c'}$, and the repaired results would not contain the secret information of the missing regions because deep hiding has the vulnerability of low redundancy and the inpainting model is trained on images that contain no secrets.

2) Pipeline: We show the complete pipeline of EBRA in Fig. 5, from which we can see that EBRA does not access the deep hiding models during the purification process. *In the erasing phase*, c' is first divided into multiple square regions. EBRA chooses some regions far away from each other and sets all corresponding pixels to 0 in each iteration. *In the repair phase*, EBRA completes all missing pixels without reconstructing the corresponding secrets. Specifically, The edge and color generators (I_1 and I_2) extract the contour and coarse color maps from c'. And then, the two generators provide feature maps of their last four layers to the inpainting model (I_3) by feature fusion. I_3 takes the incomplete image (c'^M) and the mask of missing regions as its input and repairs all missing pixels. After traversing all pixels in c', we can acquire a purified container image (i.e., $\hat{c'}$) by combining all repaired regions, from which no valid secrets can be revealed.

B. Details

1) Erasing Phase: We can remove any parts of the embedded secret image by setting pixels in the corresponding parts of the container image to 0 due to the locality vulnerability. Given that it is difficult to repair a large hole since less context is not enough to support high-quality yet precise repair, we have to erase small missing regions. However, processing small regions one by one is inefficient because the number of iterations increases as the resolution of containers increases. Therefore, we decide to erase multiple small regions simultaneously, which are far away from each other. We can repair multiple missing regions far away from each other because the influence of surrounding pixels on the repair results decays as the distance to the missing region increases.

Specifically, we select a $k \times k$ region every d regions as marked by the red boxes in c'^M in Fig. 5. And we set the pixels of all chosen regions as 0 to get the incomplete image c'^M and corresponding binary mask M. The erasing phase can be summarized as

$$M\,, c'^M = Erase(c', m, n, k, d), \tag{2}$$

where (m, n) is the start point of selecting regions (e.g. the center pixel of the first selected region). c'^M contains no secrets of the corresponding missing regions but is useless

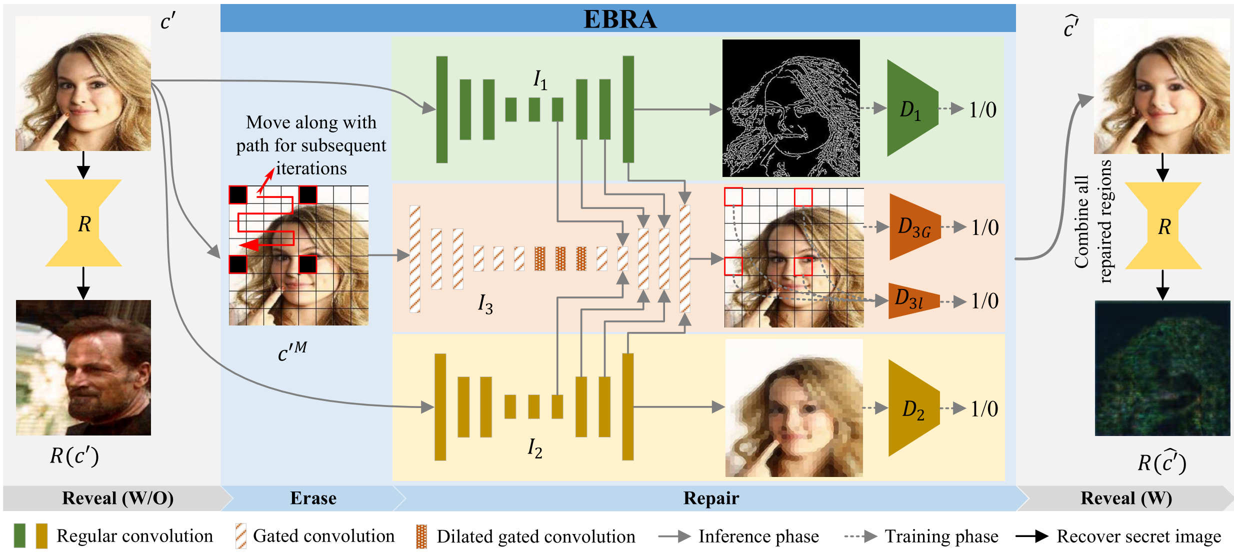


Fig. 5. Overview of EBRA. EBRA is composed of two phases: erasing and repair. The erasing phase is designed to remove embedded messages by setting all selected pixels to 0 and produces an incomplete container image. The repair phase aims to improve the usability of the incomplete image and consists of three sub-models: edge generator, color generator, and inpainting model. The edge and color generators extract the contour and coarse color maps from the complete image (i.e., c') and provide corresponding feature maps to the inpainting model. Then inpainting model completes all missing pixels based on the remaining context in c'^M and the auxiliary feature maps. The four discriminators (D_1, D_2, D_{3G}, and D_{3l}) are only considered in the training phase, which determines whether their inputs are real or fake.

since it is incomplete. So we design the following phase to improve the usability of c'^M.

2) Repair Phase: To maintain the usability of the final purified container images, we propose a quality-enhanced inpainting method to repair the missing pixels in c'^M. Given that the remaining context in c'^M alone is not enough to support precise repair, we make full use of c' to enhance the quality of the repaired images. A naive way is to feed both c' and c'^M to the inpainting model directly. But this way is risky since the inpainting model may learn to copy and paste for the repair, which brings back the removed secrets. To address this problem, we propose to extract contour and coarse color maps from c' to assist the inpainting process. Note that, *the auxiliary information we used not only is critical to precise repair but also contains little valid secret information so that no valid secrets will be leaked to the inpainting process.*

Next, we introduce how to implement our quality-enhanced inpainting and how to train involved models. We need to emphasize that we train all involved models on cover images not container images to further reduce the risk of rebuilding the missing secrets in the repair process. Therefore, we will use c instead of c' in the following descriptions.

3) Edge Generator: Lines play a crucial role in precisely delineating and defining shapes. Therefore, we propose an edge generator with an encoder-decoder structure to extract the contour map from c. In experiments, we first construct a dataset that contains images and their corresponding contour maps (e^r) by applying traditional edge detectors [49] to images. After that, we employ the adversarial loss and feature-matching loss [50] to optimize the edge generator liking

$$\min_{I_1}\left(\max_{D_1}\mathcal{L}_{1_{adv}}+\lambda_{FM}\mathcal{L}_{1_{FM}}\right), \tag{3}$$

where

$$\mathcal{L}_{1_{adv}}=\mathbb{E}\left[\log D_1(e^r)\right]+\mathbb{E}\left[\log(1-D_1(e^p))\right], \tag{4}$$

$$\mathcal{L}_{1_{FM}}=\mathbb{E}\left[\sum_{i=1}^{L_{D_1}}\frac{1}{N_{D_1}^{(i)}}\left\|D_1^{(i)}(e^r)-D_1^{(i)}(e^p)\right\|_1\right], \tag{5}$$

$$e^p=I_1(c). \tag{6}$$

λ_{FM} is a weighting factor, and we set it to 10 in experiments. L_{D_1} is the number of layers of D_1. $D_1^{(i)}$ represents the feature map in the i-th layer of D_1. $N_{D_1}^{(i)}$ is the number of elements in $D_1^{(i)}$. The adversarial loss encourages I_1 to produce realistic and clear contour images, while the feature-matching loss forces the generator to produce images having similar representations as the corresponding real images [50].

4) Color Generator: Besides the contour, we can also extract color maps from c using a color generator, which is also critical to the repair. To ensure that no valid secrets will be rebuilt during the inpainting process, we consider coarse color maps (representative color of each tiny area) rather than fine color maps (color of each pixel, i.e., the image itself). In this paper, we use the technology of super-pixel segmentation [51] to divide the whole image into irregular tiny areas which consist of pixels with similar texture, color, brightness, etc. And we fill each tiny area with its average pixel to get the ground truth of the color maps (m^r). Then, we train I_2 like what we have done to train I_1. In experiments, I_1 and I_2 share the same structure that consists of ordinary convolution layers.

5) Inpainting Model: After acquiring the useful auxiliary information, we start to repair all missing pixels using an inpainting model as

$$c^I=I_3\left(c^M, M, \left\{I_1^{(i)}(c)\right\}, \left\{I_2^{(i)}(c)\right\}\right), \tag{7}$$

where c^I is the repaired image in each iteration. c^M is the masked version of c. $\left\{I_1^{(i)}(c)\right\}$ and $\left\{I_2^{(i)}(c)\right\}$ are the sets of feature map in the i-th layer of I_1 and I_2 respectively. In experiments, we use the feature maps of the last four layers of I_1 and I_2. Using the feature maps of different layers, not the output contour and color maps, reduces the risk of rebuilding the removed secrets in the inpainting process. In experiments, we construct I_3 with gated and dilated gated convolutions, which have been proven to improve the repair ability [52], [53]. More details of the two kinds of convolutions can be found in [52].

The training of I_3 is much more complex than that of I_1 and I_2. Specifically, we train I_3 with reconstruction loss, adversarial loss, and perceptual loss, as

$$\min_{I_3}\left(\max_{D_{3_L}} \mathcal{L}_{3_{advL}} + \max_{D_{3_G}} \mathcal{L}_{3_{advG}} + \lambda_{rec}\mathcal{L}_{3_{rec}} + \lambda_{per}\mathcal{L}_{3_{per}}\right). \tag{8}$$

The reconstruction loss ($\mathcal{L}_{3_{rec}}$) is in L_1-norm form as

$$\mathcal{L}_{3_{rec}} = \|c^I - c\|_1, \tag{9}$$

which makes the c^I close to c at the pixel level. The two adversarial losses ($\mathcal{L}_{3_{advL}}$ and $\mathcal{L}_{3_{advG}}$) are used to ensure the realness of repaired images. The local adversarial loss is

$$\mathcal{L}_{3_{advL}} = \mathbb{E}\left[\log D_{3_L}(c_L)\right] + \mathbb{E}\left[\log(1 - D_{3_L}(c_L^I))\right], \tag{10}$$

where c_L^I represents the repaired region in c^I, and c_L is the corresponding local region in c. D_{3_L} is a local discriminator that determines the synthesized contents in repaired regions are real or not, which can help I_3 generate details of missing contents with sharper boundaries. The global adversarial loss is

$$\mathcal{L}_{3_{advG}} = \mathbb{E}\left[\log D_{3_G}(c)\right] + \mathbb{E}\left[\log(1 - D_{3_G}(c^I))\right], \tag{11}$$

where D_{3_G} is a global discriminator that identifies the realness of the whole repaired image. D_{3_G} makes up for the deficiency of D_{3_L} by focusing only on local regions and ignoring the consistency within and outside the masked regions. The perceptual loss ($\mathcal{L}_{3_{per}}$) is used to penalize results that are not perceptually similar to the labels, and it is defined as

$$\mathcal{L}_{3_{per}} = \mathbb{E}\left[\sum_i \frac{1}{N_{VGG}^{(i)}} \left\| VGG^{(i)}(c) - VGG^{(i)}(c^I)\right\|_1\right], \tag{12}$$

where $VGG^{(i)}$ is the feature map of the i-th layer of a pre-trained VGG19 [12] and $N_{VGG}^{(i)}$ represents the number of elements in corresponding feature map. Specifically, we follow the setting in [54] to choose the specific layers of VGG19 for calculating $\mathcal{L}_{3_{per}}$. And we set $\lambda_{rec} = 10$ and $\lambda_{per} = 1$ in experiments.

In the inference phase, we apply I_1, I_2, and I_3 to c' and c'^M as

$$c'^I = I_3\left(c'^M, M, \left\{I_1^{(i)}(c)\right\}, \left\{I_2^{(i)}(c)\right\}\right). \tag{13}$$

Then, we combine all repaired regions in repaired images to produce the final purified container image ($\hat{c'}$), and R cannot recover any valid information from $\hat{c'}$ as illustrated in Fig. 5.

TABLE III
BASELINES USED IN EVALUATIONS

Type	Methods
Common image distortion	GB, GN, JPEG, Drop, motion blurring (MB), coarse dropout (CD) [55], fancy PCA (FPCA) [56]
Defending against adversarial perturbation	pixel deflection (PD) [57], bit-depth reduction (BDR) [58]
Black-box adversarial attack	NES [41]

VI. EXPERIMENTAL EVALUATION

A. Experiment Setup

1) Dataset: Our attack is deep hiding model-agnostic and dataset-agnostic. Without losing generality, we conduct all experiments on the CelebA dataset in this paper. All images are resized to the resolution of 256×256 and the pixel values are normalized to the range of [0, 1].

2) Deep Hiding: We adopt five state-of-the-art deep hiding schemes for evaluation, including UDH[1] [15], DS[2] [14], ISGAN[3] [16], MIS[4] [59], and HCVS[5] [60]. All selected deep hiding schemes realize high-capacity data hiding. UDH, DS, and MIS are designed to hide a full-size RGB image within another one. ISGAN is designed to hide a full-size gray image with another RGB image. HCVS is designed to hide a full-sized color video within another video, and the authors of HCVS claimed that HCVS has the robustness to compression. In our experiments, we adopt all selected schemes to hide images within images. As for HCVS, each image can be regarded as a frame in a video.

3) Implementation of EBRA and Baselines: We follow the architectures of models proposed in [48] to design I_1 and I_2. Both of them are composed of down-sample layers, residual blocks, and up-sample layers. In this paper, we reduce the number of residual blocks to 3 since extracting the edge and color of the complete image is relatively easier than predicting the edge of missing regions in [48]. I_3 follows an architecture similar to the network proposed in [52], which consists of gated and dilated gated convolution layers. As for D_1 and D_2, we use the discriminator proposed in [48] to realize them. For D_{3_L} and D_{3_G}, we refer to [61] to build the two discriminators. To train I_1 and I_2, we apply Canny [49] and SLIC [51] to produce e^r and m^r, respectively. In experiments, we set $k = 50$ and $d = 2$. The code of EBRA is available.[6]

As for baselines, we mainly use box-free methods as the baselines for a fair competition. We compare EBRA with 10 baselines that can be divided into three categories (common image distortions, defending against adversarial perturbation, and adversarial attack), as shown in Table III. For common image distortions, we choose Gaussian blurring (GB), Gaussian noise (GN), JPEG compression (JPEG), dropout (Drop),

[1] https://github.com/ChaoningZhang/Universal-Deep-Hiding
[2] https://github.com/zllrunning/Deep-Steganography
[3] https://github.com/Marcovaldong/ISGAN
[4] https://github.com/m607stars/MultiImageSteganography
[5] https://github.com/muziyongshixin/pytorch-Deep-Steganography
[6] https://github.com/hcliucs/EBRA

TABLE IV
ATTACK RESULTS ON ALL BASELINES

Metric	Attack	Data hiding					Average
		UDH	DS	ISGAN	MIS	HCVS	
PSNR-C|PSNR-S	GB	31.76|11.28	32.47|10.38	32.60|6.92	32.32|20.36	32.32|11.53	32.29|12.09
	GN	26.36|11.15	26.34|5.50	26.29|4.18	26.31|10.12	26.28|11.53	26.32|8.50
	Drop	36.57|6.02	45.38|12.12	25.37|5.95	33.63|11.03	33.66|9.81	34.92|8.99
	JPEG	35.65|6.26	43.34|9.74	44.13|7.45	42.47|21.41	41.52|9.10	**41.42**|10.79
	MB	29.02|15.115	30.06|15.16	29.40|11.95	29.50|22.53	29.99|12.84	29.59|15.52
	CD	18.18|16.77	18.05|19.09	17.79|15.76	18.09|19.34	18.00|18.19	18.02|17.83
	FPCA	25.62|23.58	25.89|23.21	24.91|22.02	25.75|10.35	24.77|10.46	25.39|17.92
	PD	37.73|23.56	37.90|15.61	37.25|16.73	37.98|23.13	38.00|23.92	37.77|20.59
	BDR	23.00|9.10	22.99|8.05	23.00|6.62	22.90|10.22	23.00|10.36	22.98|8.87
	NES	27.22|10.19	27.25|5.06	27.18|4.19	27.18|10.23	27.23|11.06	27.21|8.15
	EBRA	30.75|4.54	31.47|5.74	30.94|5.21	30.04|8.10	30.16|7.96	30.67|**6.31**
VIF-C|VIF-S	GB	0.709|0.282	0.764|0.074	0.758|0.070	0.751|0.558	0.796|0.450	0.756|0.287
	GN	0.436|0.043	0.450|0.011	0.457|0.007	0.445|0.044	0.448|0.040	0.447|0.029
	Drop	0.750|0.042	0.943|0.056	0.519|0.020	0.805|0.066	0.762|0.064	0.756|0.050
	JPEG	0.842|0.065	0.888|0.014	0.889|0.017	0.878|0.469	0.932|0.166	0.886|0.146
	MB	0.441|0.259	0.497|0.224	0.478|0.167	0.468|0.411	0.509|0.267	0.479|0.266
	CD	0.772|0.745	0.771|0.726	0.773|0.692	0.772|0.763	0.769|0.729	0.771|0.731
	FPCA	0.902|0.570	0.925|0.449	0.904|0.378	0.918|0.262	0.922|0.316	**0.914**|0.395
	PD	0.852|0.417	0.854|0.198	0.848|0.230	0.857|0.444	0.855|0.427	0.853|0.343
	BDR	0.508|0.052	0.528|0.015	0.531|0.023	0.526|0.052	0.533|0.051	0.525|0.039
	NES	0.460|0.040	0.476|0.009	0.480|0.008	0.470|0.048	0.475|0.038	0.472|0.029
	EBRA	0.619|0.004	0.675|0.019	0.672|0.012	0.575|0.043	0.607|0.027	0.630|**0.021**

motion blurring (MB), coarse dropout (CD) [55], and fancy PCA (FPCA) [56]. These distortions are commonly used to study data hiding's robustness [4], [15], [62]

For baselines defending against adversarial perturbation, we choose pixel deflection (PD) [57] and bit-depth reduction (BDR) [58]. In our view, the embedded secret can be also regarded as a kind of imperceptible noise, so the removal method for adversarial perturbation may also be suitable for deep hiding. Last, we also use an adversarial attack method, NES [41], as a baseline. NES is designed to cheat classification models, so we need to modify its objective function to ensure that it can make the revealing model go wrong. The revised objective function tends to increase the L_2-norm distance between $R(c')$ and $R(\hat{c'})$. *It is worth emphasizing that NES is not a box-free way since it needs to access the revealing model through the model's input and output APIs, i.e., black-box accessing.* Therefore, NES cannot work in our box-free setting. We choose NES because the container images processed by EBRA can be regarded as a kind of adversarial example for data hiding models, and NES is still much more practical than removal attacks like that in [37].

B. Objective Evaluation

As we did in Section IV-B, we use PSNR and VIF [42] to quantify the performance of various removal attacks against different deep hiding models in our objective evaluations. It is worth emphasizing once again that there is currently no metric specifically designed for measuring the performance of removal attacks. Therefore, we rely on commonly used image quality metrics in our experiments. Moreover, previous studies [42], [45] have confirmed that VIF works better on low-quality images than PSNR, but there is a gap between VIF and human perception. As a result, we complement our objective evaluation with a subjective assessment at the end of our study.

1) Attack Basic Models: We apply EBRA and baselines to attack all five basic deep hiding models (without adversarial training), and we present the results in Table IV. The two values in each cell means (PSNR-C, PSNR-S) or (VIF-C, VIF-S), where PSNR-C, PSNR-S, VIF-C, and VIF-S represent PSNR(c',$\hat{c'}$), VIF(c',$\hat{c'}$), PSNR($R(c')$,$R(\hat{c'})$), and VIF($R(c')$,$R(\hat{c'})$), respectively. As shown in Table IV, EBRA outperforms all other methods in terms of removing hidden information while maintaining the quality of the processed container images at an acceptable level. EBRA's average PSNR-C is higher than 30, indicating the image quality is generally good. Although some baselines (e.g., JPEG, PD, and FPCA) achieve much higher PSNR-C scores, they sacrifice the removal effect significantly. For example, while the average PSNR-C score of PD is 37.77, the corresponding PSNR-S score is as high as 20.59 (the VIF-S score is 0.343). In this case, the corresponding $R(\hat{c'})$ still contains a significant amount of valid secret information, resulting in failed removal. We further demonstrate the superior performance of EBRA in balancing image quality and removal effect through visualization results shown in Fig. 6. In addition to PSNR and VIF, we also assess EBRA using alternative metrics, including SSIM, bit error rate, and pixel error rate, all of which are detailed in our supplementary materials.[7]

[7] https://arxiv.org/abs/2308.01512

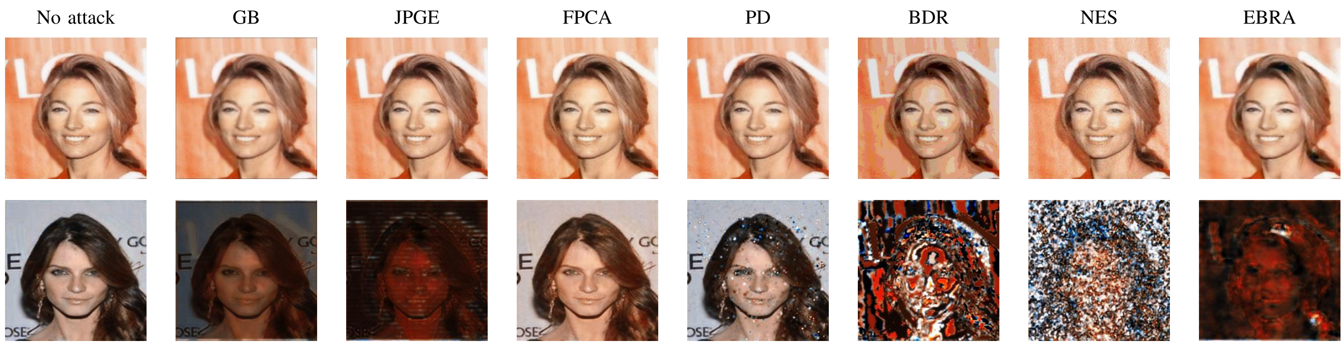


Fig. 6. Visualization examples of attacking HCVS. The images at the top show the original container image and processed container images. The images at the bottom show the corresponding recovered secret images.

TABLE V
ATTACK UDH'S ENHANCED VERSIONS

Metric	Attack	Data hiding						Average
		UDH-GB	UDH-GN	UDH-Drop	UDH-JPEG	UDH-Quan	UDH-AE	
PSNR-C|PSNR-S	GB	31.84|32.57	30.08|18.73	31.16|11.68	31.69|16.21	31.71|13.31	30.63|17.99	31.18|18.41
	GN	26.36|11.39	26.35|25.42	26.35|13.45	26.36|11.23	26.35|10.51	26.35|14.28	26.35|14.38
	Drop	37.40|8.31	32.31|14.95	36.99|26.64	37.52|7.65	38.04|9.42	33.94|11.05	36.03|13.00
	JPEG	36.52|10.40	31.20|13.41	35.46|10.14	39.80|31.14	36.38|10.63	35.87|16.07	35.87|15.30
	MB	28.91|20.92	27.30|20.79	28.42|17.73	28.72|19.27	28.94|15.84	28.27|18.91	28.43|18.91
	CD	18.14|18.04	18.11|21.86	18.11|26.08	18.12|17.90	18.11|18.31	18.11|21.00	18.12|20.53
	FPCA	25.54|25.91	25.68|33.10	26.22|27.62	25.64|25.39	25.89|26.51	25.68|29.64	25.78|28.03
	PD	37.74|23.56	37.49|37.40	37.71|25.82	37.73|23.71	37.75|23.22	37.57|27.11	**37.66**|26.80
	BDR	23.00|9.62	22.98|16.13	22.98|12.27	22.99|9.88	22.98|9.59	22.96|13.70	22.98|11.87
	NES	27.23|10.54	27.21|22.87	27.22|12.53	27.22|10.40	27.22|9.75	27.21|13.62	27.22|13.29
	EBRA	30.22|4.76	29.48|6.44	30.08|10.11	30.17|4.85	30.32|5.29	30.03|8.56	30.05|**6.67**
VIF-C|VIF-S	GB	0.710|0.657	0.611|0.299	0.673|0.150	0.695|0.507	0.701|0.123	0.616|0.352	0.668|0.348
	GN	0.436|0.040	0.430|0.334	0.434|0.066	0.437|0.047	0.436|0.037	0.432|0.073	0.434|0.099
	Drop	0.768|0.064	0.646|0.140	0.729|0.377	0.790|0.082	0.761|0.054	0.694|0.086	0.731|0.134
	JPEG	0.842|0.080	0.695|0.141	0.780|0.012	0.839|0.536	0.831|0.054	0.771|0.268	0.793|0.182
	MB	0.436|0.330	0.384|0.313	0.416|0.299	0.428|0.347	0.434|0.214	0.405|0.292	0.417|0.299
	CD	0.773|0.762	0.772|0.804	0.772|0.815	0.773|0.750	0.772|0.767	0.774|0.773	0.773|0.779
	FPCA	0.900|0.633	0.895|0.750	0.908|0.675	0.905|0.627	0.907|0.623	0.895|0.679	**0.902**|0.664
	PD	0.853|0.406	0.847|0.796	0.854|0.486	0.852|0.430	0.853|0.406	0.850|0.524	0.851|0.508
	BDR	0.506|0.055	0.505|0.285	0.504|0.077	0.507|0.060	0.507|0.050	0.501|0.108	0.505|0.106
	NES	0.460|0.040	0.453|0.332	0.458|0.066	0.462|0.045	0.460|0.036	0.457|0.075	0.458|0.099
	EBRA	0.560|0.012	0.493|0.021	0.544|0.016	0.523|0.008	0.546|0.013	0.563|0.022	0.538|**0.015**

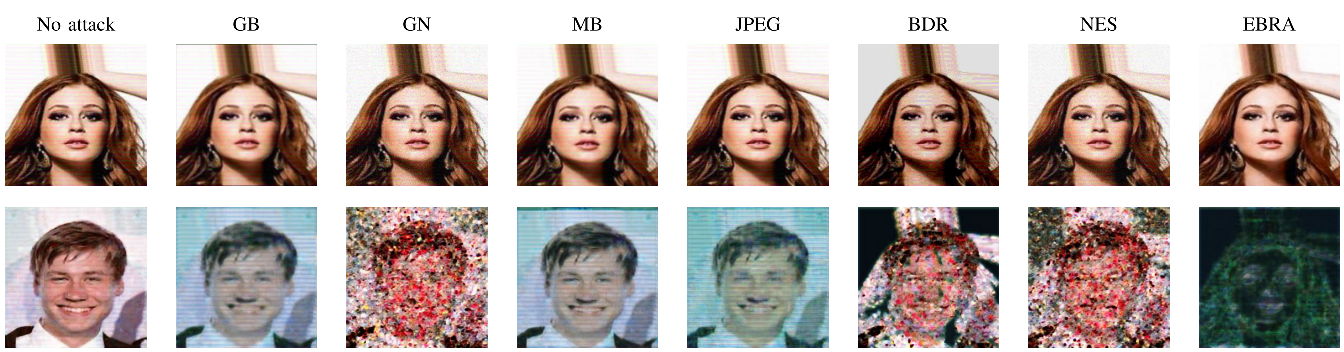


Fig. 7. Visualization examples of attacking UDH-AE. The images at the top show the original container image and processed container images. The images at the bottom show the corresponding recovered secret images.

2) Attack Robust Models: We also consider robust-enhanced models by employing adversarial training. Specifically, we choose UDH, DS, and ISGAN as the basic methods, and then use GB, GN, Drop, JPEG, quantification (Quan), and a pre-trained auto-encoder (AE) as the potential distortion. Following the configurations in [4] and [15] that adds different noise layers between the hiding and revealing models to simulate possible distortions, we get the corresponding

TABLE VI

ATTACK RESULTS ON ISGAN'S' ENHANCED VERSIONS

<table>
<tr><th rowspan="2">Metric</th><th rowspan="2">Attack</th><th colspan="6">Data hiding</th><th rowspan="2">Average</th></tr>
<tr><th>DS-GB</th><th>DS-GN</th><th>DS-Drop</th><th>DS-JPEG</th><th>DS-Quan</th><th>DS-AE</th></tr>
<tr><td rowspan="11">PSNR-C|PSNR-S</td><td>GB</td><td>32.33|35.67</td><td>28.89|11.33</td><td>30.60|16.29</td><td>32.02|22.09</td><td>31.89|10.66</td><td>31.86|12.62</td><td>31.27|18.11</td></tr>
<tr><td>GN</td><td>26.31|7.18</td><td>26.30|27.00</td><td>26.34|10.78</td><td>26.29|11.75</td><td>26.34|10.37</td><td>26.27|11.84</td><td>26.31|13.15</td></tr>
<tr><td>Drop</td><td>40.58|14.23</td><td>32.22|14.55</td><td>34.86|28.45</td><td>33.05|11.09</td><td>37.79|13.51</td><td>30.70|12.12</td><td>34.87|15.66</td></tr>
<tr><td>JPEG</td><td>41.40|10.58</td><td>31.12|11.16</td><td>34.45|11.13</td><td>40.56|29.44</td><td>39.47|9.63</td><td>39.43|19.64</td><td>37.74|15.26</td></tr>
<tr><td>MB</td><td>29.97|15.12</td><td>27.02|15.85</td><td>28.22|19.19</td><td>29.68|18.09</td><td>29.22|15.32</td><td>29.81|16.86</td><td>28.99|16.74</td></tr>
<tr><td>CD</td><td>18.11|21.69</td><td>18.11|24.58</td><td>18.06|27.19</td><td>18.03|21.70</td><td>18.08|20.77</td><td>18.15|19.79</td><td>18.09|22.62</td></tr>
<tr><td>FPCA</td><td>25.82|26.33</td><td>25.90|33.66</td><td>25.86|30.53</td><td>25.36|12.36</td><td>26.08|27.51</td><td>25.83|18.26</td><td>25.81|24.77</td></tr>
<tr><td>PD</td><td>37.98|18.76</td><td>37.64|37.42</td><td>37.62|22.80</td><td>37.94|19.72</td><td>37.75|23.07</td><td>37.95|26.26</td><td>37.81|24.67</td></tr>
<tr><td>BDR</td><td>23.00|9.86</td><td>22.94|17.22</td><td>23.00|13.94</td><td>22.96|10.03</td><td>23.07|11.48</td><td>22.99|14.25</td><td>22.99|12.80</td></tr>
<tr><td>NES</td><td>27.24|6.90</td><td>27.22|25.65</td><td>27.24|10.88</td><td>27.22|9.47</td><td>27.27|9.93</td><td>27.20|11.68</td><td>27.23|12.42</td></tr>
<tr><td>EBRA</td><td>30.33|8.55</td><td>27.55|11.12</td><td>29.08|10.71</td><td>30.20|11.03</td><td>30.13|10.06</td><td>30.19|11.12</td><td>29.58|10.43</td></tr>
<tr><td rowspan="11">VIF-C|VIF-S</td><td>GB</td><td>0.744|0.779</td><td>0.583|0.042</td><td>0.642|0.274</td><td>0.735|0.645</td><td>0.701|0.093</td><td>0.661|0.189</td><td>0.678|0.337</td></tr>
<tr><td>GN</td><td>0.445|0.016</td><td>0.427|0.352</td><td>0.436|0.087</td><td>0.444|0.023</td><td>0.447|0.025</td><td>0.432|0.049</td><td>0.438|0.092</td></tr>
<tr><td>Drop</td><td>0.880|0.076</td><td>0.685|0.221</td><td>0.739|0.406</td><td>0.784|0.058</td><td>0.807|0.091</td><td>0.581|0.064</td><td>0.746|0.153</td></tr>
<tr><td>JPEG</td><td>0.863|0.020</td><td>0.632|0.017</td><td>0.710|0.019</td><td>0.854|0.614</td><td>0.807|0.023</td><td>0.812|0.325</td><td>0.780|0.170</td></tr>
<tr><td>MB</td><td>0.485|0.233</td><td>0.396|0.221</td><td>0.426|0.251</td><td>0.479|0.421</td><td>0.454|0.218</td><td>0.442|0.259</td><td>0.447|0.267</td></tr>
<tr><td>CD</td><td>0.770|0.750</td><td>0.770|0.804</td><td>0.773|0.825</td><td>0.770|0.766</td><td>0.773|0.758</td><td>0.772|0.731</td><td>0.771|0.772</td></tr>
<tr><td>FPCA</td><td>0.927|0.603</td><td>0.911|0.818</td><td>0.911|0.746</td><td>0.925|0.256</td><td>0.922|0.674</td><td>0.919|0.436</td><td>0.919|0.589</td></tr>
<tr><td>PD</td><td>0.855|0.287</td><td>0.853|0.782</td><td>0.853|0.445</td><td>0.855|0.303</td><td>0.853|0.407</td><td>0.858|0.481</td><td>0.854|0.451</td></tr>
<tr><td>BDR</td><td>0.526|0.032</td><td>0.512|0.276</td><td>0.517|0.127</td><td>0.525|0.048</td><td>0.526|0.049</td><td>0.510|0.097</td><td>0.519|0.105</td></tr>
<tr><td>NES</td><td>0.472|0.014</td><td>0.453|0.343</td><td>0.461|0.095</td><td>0.471|0.035</td><td>0.471|0.025</td><td>0.457|0.054</td><td>0.464|0.094</td></tr>
<tr><td>EBRA</td><td>0.582|0.019</td><td>0.532|0.011</td><td>0.519|0.012</td><td>0.621|0.020</td><td>0.564|0.021</td><td>0.523|0.041</td><td>0.557|0.021</td></tr>
</table>

TABLE VII

ATTACK RESULTS ON ISGAN'S' ENHANCED VERSIONS

<table>
<tr><th rowspan="2">Metric</th><th rowspan="2">Attack</th><th colspan="6">Data hiding</th><th rowspan="2">Average</th></tr>
<tr><th>ISGAN-GB</th><th>ISGAN-GN</th><th>ISGAN-Drop</th><th>ISGAN-JPEG</th><th>ISGAN-Quan</th><th>ISGAN-AE</th></tr>
<tr><td rowspan="11">PSNR-C|PSNR-S</td><td>GB</td><td>32.74|29.38</td><td>30.50|13.43</td><td>32.47|10.28</td><td>32.59|10.54</td><td>32.18|6.86</td><td>32.29|12.36</td><td>32.13|13.81</td></tr>
<tr><td>GN</td><td>26.24|8.27</td><td>26.29|18.24</td><td>26.29|9.16</td><td>26.22|7.05</td><td>26.20|7.34</td><td>26.29|9.93</td><td>26.25|10.00</td></tr>
<tr><td>Drop</td><td>23.56|8.65</td><td>20.55|11.85</td><td>23.79|9.49</td><td>22.97|7.59</td><td>21.60|7.84</td><td>25.00|9.26</td><td>22.91|9.11</td></tr>
<tr><td>JPEG</td><td>43.86|11.33</td><td>36.35|14.87</td><td>42.54|11.65</td><td>44.91|30.74</td><td>40.46|8.65</td><td>42.44|15.02</td><td>41.76|15.38</td></tr>
<tr><td>MB</td><td>29.90|12.68</td><td>27.45|19.21</td><td>29.44|12.84</td><td>29.76|15.54</td><td>28.81|14.12</td><td>29.51|17.77</td><td>29.14|15.36</td></tr>
<tr><td>CD</td><td>17.65|16.19</td><td>17.77|21.22</td><td>18.18|20.44</td><td>17.72|15.91</td><td>17.72|19.73</td><td>17.65|17.09</td><td>17.78|18.43</td></tr>
<tr><td>FPCA</td><td>25.21|26.56</td><td>24.68|20.44</td><td>24.79|19.90</td><td>25.04|26.53</td><td>24.67|26.78</td><td>24.33|31.07</td><td>24.79|25.21</td></tr>
<tr><td>PD</td><td>37.38|20.18</td><td>36.85|30.63</td><td>37.35|18.00</td><td>37.47|20.55</td><td>37.15|19.72</td><td>37.31|26.06</td><td>37.25|22.52</td></tr>
<tr><td>BDR</td><td>22.96|9.84</td><td>23.01|14.79</td><td>22.99|9.96</td><td>22.90|9.64</td><td>22.81|9.58</td><td>22.99|12.89</td><td>22.94|11.12</td></tr>
<tr><td>NES</td><td>27.16|7.57</td><td>27.20|17.72</td><td>27.20|9.05</td><td>27.13|10.85</td><td>27.10|7.29</td><td>27.19|9.32</td><td>27.16|10.30</td></tr>
<tr><td>EBRA</td><td>30.67|9.28</td><td>29.80|10.75</td><td>30.40|9.20</td><td>29.10|7.02</td><td>28.61|8.22</td><td>30.29|10.42</td><td>29.81|9.15</td></tr>
<tr><td rowspan="11">VIF-C|VIF-S</td><td>GB</td><td>0.763|0.601</td><td>0.645|0.212</td><td>0.769|0.198</td><td>0.774|0.162</td><td>0.720|0.044</td><td>0.739|0.146</td><td>0.735|0.227</td></tr>
<tr><td>GN</td><td>0.453|0.016</td><td>0.462|0.201</td><td>0.454|0.039</td><td>0.451|0.023</td><td>0.454|0.014</td><td>0.454|0.040</td><td>0.455|0.056</td></tr>
<tr><td>Drop</td><td>0.437|0.017</td><td>0.346|0.051</td><td>0.451|0.089</td><td>0.412|0.019</td><td>0.386|0.018</td><td>0.486|0.033</td><td>0.420|0.038</td></tr>
<tr><td>JPEG</td><td>0.889|0.043</td><td>0.733|0.153</td><td>0.896|0.057</td><td>0.914|0.556</td><td>0.832|0.047</td><td>0.853|0.198</td><td>0.853|0.176</td></tr>
<tr><td>MB</td><td>0.493|0.158</td><td>0.405|0.278</td><td>0.489|0.174</td><td>0.480|0.237</td><td>0.452|0.235</td><td>0.468|0.259</td><td>0.464|0.224</td></tr>
<tr><td>CD</td><td>0.772|0.688</td><td>0.774|0.755</td><td>0.776|0.735</td><td>0.772|0.675</td><td>0.771|0.734</td><td>0.774|0.707</td><td>0.773|0.716</td></tr>
<tr><td>FPCA</td><td>0.920|0.529</td><td>0.897|0.458</td><td>0.910|0.287</td><td>0.922|0.599</td><td>0.922|0.609</td><td>0.904|0.741</td><td>0.913|0.537</td></tr>
<tr><td>PD</td><td>0.850|0.299</td><td>0.844|0.588</td><td>0.850|0.207</td><td>0.851|0.372</td><td>0.848|0.348</td><td>0.849|0.567</td><td>0.849|0.397</td></tr>
<tr><td>BDR</td><td>0.524|0.038</td><td>0.532|0.151</td><td>0.522|0.046</td><td>0.526|0.048</td><td>0.531|0.043</td><td>0.527|0.079</td><td>0.527|0.067</td></tr>
<tr><td>NES</td><td>0.478|0.014</td><td>0.486|0.203</td><td>0.478|0.039</td><td>0.475|0.023</td><td>0.478|0.015</td><td>0.479|0.041</td><td>0.479|0.056</td></tr>
<tr><td>EBRA</td><td>0.594|0.016</td><td>0.520|0.042</td><td>0.591|0.027</td><td>0.588|0.021</td><td>0.576|0.023</td><td>0.651|0.028</td><td>0.587|0.026</td></tr>
</table>

robust-enhanced models. For example, UDH-GB means a UDH model that is adversarially trained with GB. We show all results on these robust models in V, VI, and VII.

First, EBRA continues to show its superiority in completely removing the embedded secret images while maintaining the quality of processed container images at an acceptable level (with an average PSNR-C above 29). Despite the average PSNR-C and VIF-C scores of EBRA dropping slightly facing these robust models, this is understandable. In Fig. 7, UDH-AE tends to produce stripes in the container images, which are retained in the processed container images produced by most removal attacks except for EBRA. The removal of stripes by EBRA inevitably increases the distance between the container images before and after attacking and

TABLE VIII
EVALUATION ON MORE ROBUST MODELS

Attack	PSNR-C|PSNR-S		VIF-C|VIF-S	
	UDH-C	UDH-EBRA	UDH-C	UDH-EBRA
EBRA	28.31|10.67	29.43|16.81	0.530|0.015	0.569|0.157
EBRA*	-	28.92|11.97	-	0.492|0.036

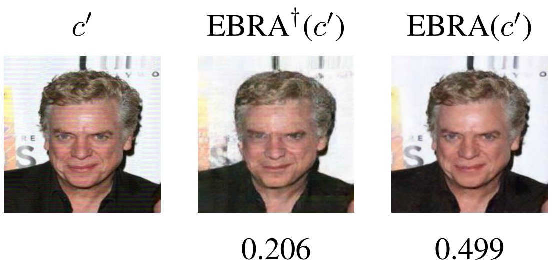


Fig. 8. Comparison between EBRA and EBRA† in quality. The values mean VIF-C scores.

TABLE IX
EVALUATIONS OF EBRA†

Model	PSNR-C|PSNR-S	VIF-C|VIF-S
UDH	25.44|4.57	0.302|0.004
DS	26.03|9.79	0.349|0.017
ISGAN	25.55|8.61	0.341|0.020
MIS	25.79|8.21	0.325|0.041
HCVS	25.67|8.84	0.348|0.038

is also the key to completely removing the embedded secret image.

Second, we observe that adversarial training does improve the robustness against known distortions significantly, and *this improvement shows some degree of generalization to unknown distortions*. For example, in Table V, although GN poses a great threat to UDH (the PSNR-S and VIF-S scores are 11.15 and 0.043, respectively), it cannot attack UDH-GN (the PSNR-S and VIF-S scores increase to 25.42 and 0.334 respectively). Additionally, UDH-GN is also resistant to the Drop attack, despite though Drop seriously threatens UDH.

Besides, we adopt two approaches to further enhance the robustness against EBRA. First, following the experimental setting in [15], we combine different distortions (e.g. GB, GN, Drop, and JPEG) simultaneously in the adversarial training and train a combined UDH (UDH-C) like that in [15]. Second, we conduct targeted adversarial training to get UDH-EBRA by assuming that the defenders have knowledge of the details of EBRA. The results of attacking the two models using EBRA are shown in Table VIII. It is clear to see that EBRA still works against UDH-C but fails to attack UDH-EBRA (the VIF-S increases to 0.157 and the PSNR-S is 16.81). So, targeted adversarial training is still effective. However, it relies on a strong assumption and seems to be impractical in the real world. Additionally, we find that *the effectiveness of targeted adversarial training is fragile when some specific parameter settings are changed*. To confirm this, we train a new group

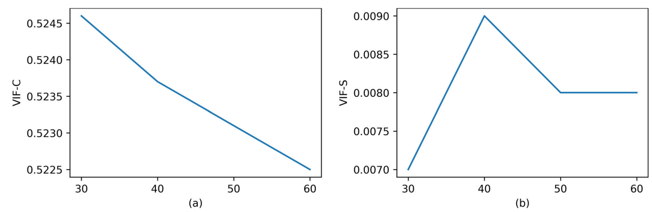


Fig. 9. Evaluation results of different k.

of I_1, I_2, and I_3 by adjusting hyperparameters, such as the initialization of model parameters, learning rate, k, and the number of convolution kernels in hidden layers (randomly reduce z convolution kernels, where $z = 0, 1, 2$). After that, we obtain a new EBRA, denoted as EBRA*. Experimental results (Table VIII) show that EBRA* significantly reduces the VIF-S scores. To sum up, the threat of adversarial training to our EBRA is limited, and this encourages us to find more effective methods to resist EBRA in our future research.

C. Ablation Study

We conduct an ablation study to verify the importance of contour and color information, in which we only consider I_3 to build EBRA (denoted as EBRA†). The experimental results are shown in Table IX. After removing I_1 and I_2, the PNSR-C and VIF-C decrease significantly. Fig. 8 also confirms this. As we can see in Fig. 8, the right image contains clearer background (e.g., the letter "S"), eyes, and hair than the middle one. All evidence reflects the superiority of our quality-enhanced inpainting. At the same time, the auxiliary feature maps we used do not provide an advantage for the secret image revealing since the PSNR-S and VIS-S scores are close before and after using I_1 and I_2.

D. Evaluation on Different k

In experiments, we also adopt different k and find that k has a small effect on the removal effect when k is relatively small. Specifically, we train I_3 by setting $k = 30, 40, 50, 60$ respectively. The corresponding VIF-C and VIF-S results are presented in Fig. 9. As shown in Fig. 9, the VIF-C scores (or VIF-S scores) are close. A recent work [46] poses the possibility of repairing large missing regions. However, as shown in [46], the semantic information of repaired contents differs greatly from the original's when the missing region is too large, although the repair pixels are natural. To maintain the semantic information in c', we do not consider large-scale erasing in a single step and randomly choose $k = 50$.

E. Efficiency

We test the efficiency of different removal attacks and a famous steganalysis method SRNet [28] on a machine. Specifically, we use these methods to process 300 container images and calculate the average time. Experiment results are shown in Table X. It's worth noting that, we use I_1 and I_2 only once for each container. And with $d = 2$, EBRA needs only 9 cycles to process a container image using I_3 if the batch size is 1. However, accessing I_3 step by step

TABLE X
THE TIME OVERHEAD (MS) OF DIFFERENT METHODS REQUIRED TO PROCESS AN IMAGE

Removal attack												Steganalysis
GB	GN	Drop	JPEG	MB	CD	FPCA	PD	BDR	NES	EBRA (batch size=1)	EBRA (batch size=9)	SRNet
0.027	1.58	2.12	1.34	2.98	1.85	5.19	9.39	1.54	463.72	39.87	7.75	4.39

TABLE XI
FAILURE RATE OF ALL TEST METHODS IN THE SUBJECTIVE EVALUATION

Attack	UDH	UHD-GB	UDH-GN	UDH-Drop	UDH-JPEG	UDH-Quan	UDH-AE	Average
GB	100%	100%	100%	100%	100%	100%	100%	100%
GN	12%	16%	100%	63%	30%	86%	98%	58%
Drop	11%	93%	100%	100%	97%	45%	99%	78%
JPEG	97%	100%	100%	4%	100%	100%	100%	86%
MB	88%	99%	99%	100%	98%	97%	100%	97%
CD	100%	100%	100%	100%	100%	100%	100%	100%
FPCA	99%	100%	100%	100%	100%	100%	100%	100%
PD	100%	100%	100%	100%	100%	100%	100%	100%
BDR	26%	46%	100%	65%	25%	92%	100%	65%
NES	9%	13%	100%	44%	26%	88%	99%	54%
EBRA	0%	0%	0%	0%	2%	1%	3%	**1**%

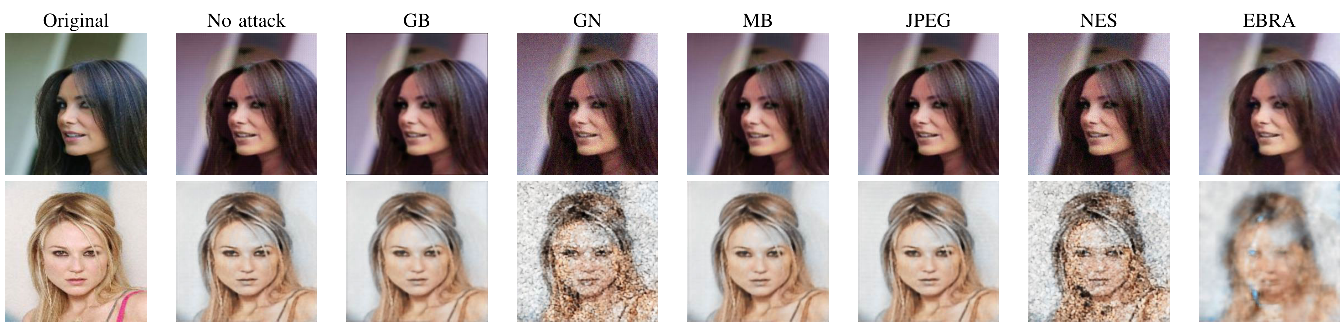


Fig. 10. Remove secret images from significantly distorted container images. The images at the top, from left to right, show the original cover image, the original container image, and the processed container images. The images at the bottom display the original secret image and its corresponding recovered secret images.

(batch size is 1) increases the number of interactions between system memory and GPU memory so the time overhead of EBRA is not satisfactory. To further reduce the time overhead, we prepare all 9 incomplete container images and feed them together to I_3 as a batch (i.e., the batch size is 9 and the number of interactions decreases to 1). Finally, we reduce the time overhead from 39.87 ms to 7.75 ms which is perfectly acceptable in most cases.

F. Subjective Evaluation

As we have previously claimed, there is no metric designed specifically for evaluating removal attacks. Therefore, objective results measured by popular image quality metrics may not fully reflect the real removal effect. To address this issue, we conduct a subjective evaluation. In this evaluation, we invite 50 observers to determine whether valid information (e.g., hair, eye, glasses, hat, etc.) can be observed in $R(\hat{c'})$. If the answer is no, the corresponding removal is successful. Otherwise, the removal fails. All of these observers are college students majoring in science, engineering, language, and biology; none of them has background knowledge of data hiding. They all have normal or corrected vision, aged from 20 to 30 years old. For each data hiding method, we prepare 100 pairs of $(R(c'), R(\hat{c'}))$, and then ask all observers to browse every pair of images. We collected all feedback and calculated the average removal failure rate as shown in Table XI. A lower removal failure rate indicates a better secret removal effect. We can clearly observe that EBRA achieves an extremely low failure rate in Table XI, averaging only 1%. Besides, we also get an interesting phenomenon: *adopting GN or AE in the adversarial training can better improve the robustness than using other adversarial distortions* because the corresponding models show the best generalization to unknown distortions.

VII. LIMITATION

In the preceding section, we have confirmed that when hiding is imperceptible (i.e., $\mathbb{E}_{\sim s,c} D(c, c') \leq \xi_1$), EBRA can effectively remove secret images from the container images. However, when hiding becomes perceptible (i.e., $\mathbb{E}_{\sim s,c} D(c, c') \geq \xi_1$), the performance of EBRA might be affected. This is primarily due to EBRA possibly mistaking the severe distortions caused by the perceptible hiding as inherent properties of the container image, thereby attempting to recover these distortions. To substantiate this, we adversarially train HCVS [60] with an auto-encoder while controlling the PSNR(c,c') at a low level (PSNR(c,c')=26.9) to ensure that the hiding would result in severe distortions in the container images. As we illustrate in Fig. 10, HCVS-AE seems to produce the container image by adding a purple filter to the cover image. And this color cast across the entire image scope records valid secret information and makes EBRA perceive it as an inherent property of the container image itself. As a result, this color cast is well-preserved in the output of EBRA, causing the secret image to not be completely removed from the container image.

Nevertheless, EBRA still maintains the recovered secret image at a low level of recognizability. In contrast, the recovered secret images corresponding to other removal methods are much more recognizable and are not even affected by the removal attacks (e.g., GB, MB, and JPEG). Lastly, we must argue that although EBRA has this limitation, it remains applicable to attacking current mainstream deep hiding schemes, as them [14], [15], [16], [19], [20], [21], [59], [60] still focus on imperceptible hiding.

VIII. Conclusion

In this paper, we challenge the robustness of existing deep hiding schemes. To this end, we first explore the vulnerabilities of current deep hiding schemes. Based on our observations, we design a two-phase box-free removal attack. To better maintain the usability of purified container images, we design two auxiliary networks to extract the contour and coarse color maps from the container images and transmit the extracted features to the inpainting model through feature fusion. The subjective and objective experimental results reflect that our EBRA can remove embedded secret images thoroughly, even the deep hiding models are enhanced by adversarial training. We hope our work can heat up the arms race to inspire the designs of more advanced deep hiding schemes and attacks in the future.

References

[1] C. Zhang, C. Lin, P. Benz, K. Chen, W. Zhang, and I. S. Kweon, "A brief survey on deep learning based data hiding, steganography and watermarking," 2021, *arXiv:2013.01607*, 2021.

[2] J. Sun, G. Xu, T. Zhang, X. Yang, M. Alazab, and R. H. Deng, "Verifiable, fair and privacy-preserving broadcast authorization for flexible data sharing in clouds," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 683–698, 2023.

[3] J. Zhang et al., "Deep model intellectual property protection via deep watermarking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4005–4020, Aug. 2022.

[4] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "HiDDeN: Hiding data with deep networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 657–672.

[5] T. Pevnỳ, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Information Hiding*. Berlin, Germany: Springer, 2010, pp. 161–177.

[6] B. Li, S. Tan, M. Wang, and J. Huang, "Investigation on cost assignment in spatial image steganography," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 8, pp. 1264–1277, Aug. 2014.

[7] 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.

[8] B. Isac and V. Santhi, "A study on digital image and video watermarking schemes using neural networks," *Int. J. Comput. Appl.*, vol. 12, no. 9, pp. 1–6, Jan. 2011.

[9] H. Kandi, D. Mishra, and S. R. K. S. Gorthi, "Exploring the learning capabilities of convolutional neural networks for robust image watermarking," *Comput. Secur.*, vol. 65, pp. 247–268, Mar. 2017.

[10] S.-M. Mun, S.-H. Nam, H.-U. Jang, D. Kim, and H.-K. Lee, "A robust blind watermarking using convolutional neural network," 2017, *arXiv:1704.03248*.

[11] T. Xiang, Y. Yang, S. Guo, H. Liu, and H. Liu, "PRNet: A progressive recovery network for revealing perceptually encrypted images," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 3537–3545.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–10.

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[14] S. Baluja, "Hiding images in plain sight: Deep steganography," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 2066–2076.

[15] C. Zhang, P. Benz, A. Karjauv, G. Sun, and I. S. Kweon, "UDH: Universal deep hiding for steganography, watermarking, and light field messaging," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 10223–10234.

[16] R. Zhang, S. Dong, and J. Liu, "Invisible steganography via generative adversarial networks," *Multimedia Tools Appl.*, vol. 78, no. 7, pp. 8559–8575, Apr. 2019.

[17] C. Zhang, A. Karjauv, P. Benz, and I. S. Kweon, "Towards robust deep hiding under non-differentiable distortions for practical blind watermarking," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 5158–5166.

[18] D. Jung, H. Bae, H.-S. Choi, and S. Yoon, "PixelSteganalysis: Pixel-wise hidden information removal with low visual degradation," *IEEE Trans. Depend. Secure Comput.*, vol. 20, no. 1, pp. 331–342, Jan. 2023.

[19] C. Yu, "Attention based data hiding with generative adversarial networks," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 34, 2020, pp. 1120–1128.

[20] S.-P. Lu, R. Wang, T. Zhong, and P. L. Rosin, "Large-capacity image steganography based on invertible neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10811–10820.

[21] Z. Guan et al., "DeepMIH: Deep invertible network for multiple image hiding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 372–390, Jan. 2023.

[22] Y. Xu, C. Mou, Y. Hu, J. Xie, and J. Zhang, "Robust invertible image steganography," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7865–7874.

[23] X. Luo, R. Zhan, H. Chang, F. Yang, and P. Milanfar, "Distortion agnostic deep watermarking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13545–13554.

[24] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using image quality metrics," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 221–229, Feb. 2003.

[25] J. Kodovsky, 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.

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

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

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

[29] W. You, H. Zhang, and X. Zhao, "A Siamese CNN for image steganalysis," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 291–306, 2021.

[30] F. Li, Y. Zeng, X. Zhang, and C. Qin, "Ensemble stego selection for enhancing image steganography," *IEEE Signal Process. Lett.*, vol. 29, pp. 702–706, 2022.

[31] J. Hayes and G. Danezis, "Generating steganographic images via adversarial training," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, I. Guyon et al., Eds., vol. 30. Red Hook, NY, USA: Curran Associates, 2017, pp. 1951–1960.

[32] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang, "CNN-based adversarial embedding for image steganography," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 8, pp. 2074–2087, Aug. 2019.

[33] G. Paul and I. Mukherjee, "Image sterilization to prevent LSB-based steganographic transmission," 2010, *arXiv:1012.5573*.

[34] M. Imon, P. Goutam, and A. J. Jawahar, "Defeating steganography with multibit sterilization using pixel eccentricity," *IPSI BgD Trans. Adv. Res.*, vol. 11, no. 1, pp. 25–34, Jan. 2015.

[35] D.-C. Wu and W.-H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognit. Lett.*, vol. 24, nos. 9–10, pp. 1613–1626, Jun. 2003.

[36] S. Ganguly and I. Mukherjee, "Image sterilization through adaptive noise blending in integer wavelet transformation," in *Proc. IEEE 19th India Council Int. Conf. (INDICON)*, Nov. 2022, pp. 1–6.

[37] I. Corley, J. Lwowski, and J. Hoffman, "Destruction of image steganography using generative adversarial networks," 2019, *arXiv:1912.10070*.

[38] S. Y. Ameen and M. R. Al-Badrany, "Optimal image steganography content destruction techniques," in *Proc. Int. Conf. Syst., Control, Signal Process. Informat. (SCSPI)*, 2013, pp. 453–457.

[39] P. P. Amritha, M. Sethumadhavan, R. Krishnan, and S. K. Pal, "Anti-forensic approach to remove stego content from images and videos," *J. Cyber Secur. Mobility*, vol. 8, pp. 295–320, Apr. 2018.

[40] T. Xiang, H. Liu, S. Guo, Y. Gan, and X. Liao, "EGM: An efficient generative model for unrestricted adversarial examples," *ACM Trans. Sensor Netw.*, vol. 18, no. 4, pp. 1–25, Nov. 2022.

[41] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 2137–2146.

[42] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Feb. 2006.

[43] T. Xiang, S. Guo, and X. Li, "Perceptual visual security index based on edge and texture similarities," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 5, pp. 951–963, May 2016.

[44] H. Hofbauer and A. Uhl, "Visual quality indices and lowquality images," in *Proc. 2nd Eur. Workshop Vis. Inf. Process. (EUVIP)*, Jul. 2010, pp. 171–176.

[45] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.

[46] S. Zhao et al., "Large scale image completion via co-modulated generative adversarial networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–14.

[47] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–14, Aug. 2017.

[48] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, "Edge-Connect: Generative image inpainting with adversarial edge learning," 2019, *arXiv:1901.00212*.

[49] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[50] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8798–8807.

[51] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[52] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, "Free-form image inpainting with gated convolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4470–4479.

[53] Y. Jo and J. Park, "SC-FEGAN: Face editing generative adversarial network with user's sketch and color," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1745–1753.

[54] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 694–711.

[55] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*.

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 25, 2012, pp. 1106–1114.

[57] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer, "Deflecting adversarial attacks with pixel deflection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8571–8580.

[58] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," 2017, *arXiv:1704.01155*.

[59] A. Das, J. S. Wahi, M. Anand, and Y. Rana, "Multi-image steganography using deep neural networks," 2021, *arXiv:2101.00350*.

[60] X. Weng, Y. Li, L. Chi, and Y. Mu, "High-capacity convolutional video steganography with temporal residual modeling," in *Proc. Int. Conf. Multimedia Retr.*, Jun. 2019, pp. 87–95.

[61] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5505–5514.

[62] L. Meng, X. Jiang, Z. Zhang, Z. Li, and T. Sun, "A robust coverless image steganography based on an end-to-end hash generation model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 7, pp. 3542–3558, Jul. 2023.