Fast and Effective Overwrite Attack Against DNN-based Image Watermarking Models

Shaoxin Li, Xiaofeng Liao, *Fellow, IEEE,*, Qiqi Zhang, Yuanqi Xue and Lingyang Chu

***Abstract*—Deep neural network (DNN)-based image watermarking models have been widely recognized as an effective way to manage the huge amount of AI-generated images. However, the vulnerability of such models to different forms of adversarial attacks has been a critical concern. Among the existing forms of attacks in the literature, image-dependent attacks cannot launch real-time attacks on a large number of watermarked images, because they need to train a new noise image to attack each new watermarked image; image-regeneration attacks either require a lot of information about the watermarking system or cause too much damage to the attacked image. To fill the gap in the existing forms of attacks, in this paper, we propose a novel form of attack named "fast and effective overwrite attack (FEOA)", which achieves an extremely fast attack speed and strong attack effectiveness. In particular, we discovered a single noise image, when directly added to many watermarked images, can overwrite their true watermark messages to different ones in milliseconds. We also develop an adaptive version of FEOA, which trains k different noise images and applies the principle of divide and conquer to significantly improve attack effectiveness. Our work opens the door to quickly launching massive overwrite attacks on a large number of watermarked images, revealing a new robustness issue of DNN-based image watermarking models. Extensive experiments demonstrate the outstanding attack time efficiency and effectiveness of our methods.**

***Index Terms*—Image watermarking, deep neural networks, overwrite attack.**

I. Introduction

TO effectively manage the rapidly growing number of AI-generated images, many deep neural network (DNN)-based image watermarking models [1]–[16] have been developed to enable image attribution [14], [17], establish proof of ownership [18]–[20], implement copy controls [21], and achieve authentication purposes [22]. As these models become widely deployed, it is increasingly important to assess their robustness against *adversarial attacks*, which typically manipulate watermarked images in a way that disrupts the watermark extraction process or misleads the authentication mechanism. Studying these attacks not only reveals inherent vulnerabilities in DNN-based image watermarking models but also drives the development of more resilient models, ultimately safeguarding the integrity of AI-generated images.

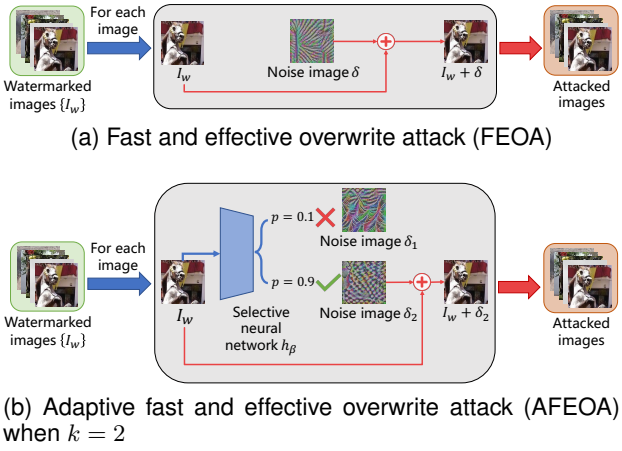


Fig. 1. The overview diagrams of FEOA and AFEOA. (a) FEOA overwrites the message of each watermarked image by adding a noise image δ to it. (b) AFEOA overwrites the message of each watermarked image by adding a noise image that is chosen from a set of k noise images $\{\delta_1, \ldots, \delta_k\}$ by the selective neural network h_β.

Some recent research [23]–[27] has revealed the vulnerability of DNN-based image watermarking models to different forms of adversarial attacks. As discussed later in Section II, *image-dependent attacks* [23], [25], [26], [28], [29] train a unique noise image tailored for each watermarked image to remove or overwrite its watermark. While often effective, these methods require significant computational time per attack, which prohibits them from launching real-time attacks on a large amount of watermarked images. On the other hand, *image-regeneration attacks* [24]–[27], [30], [31] remove the watermark of a watermarked image by regenerating a visually similar image without the watermark. These attacks are generally faster than image-dependent attacks. Nevertheless, they either demand extensive prior knowledge about the watermarking system [24], [30] or cause visible damage to the image quality due to uncontrolled overly smoothing on watermarked images [26], [27], [31].

To the best of our knowledge, how to launch fast and effective overwrite attacks against DNN-based image watermarking models is a novel and challenging problem that has not been systematically studied in the literature [23]–[27]. Bridging this

This work was done by Shaoxin Li during his visit at McMaster University when supervised by Lingyang Chu. This work is supported in part by the NSERC Discovery Grant program (RGPIN-2022-04977), in part by National Natural Science Foundation of China (Grant no. 61932006), in part by the Natural Science Foundation of Chongqing (Innovation and Development Joint Fund) under grant CSTB2023NSCQ-LZX0149, in part by the Fundamental Research Funds for the Central Universities under grant 2023CDJKYJH019, and in part by the scholarship from China Scholarship Council. (*Corresponding author: Xiaofeng Liao*)

S. Li and X. Liao are with the College of Computer Science, Chongqing University, Chongqing, 400044, China (email: {shaoxin.li, xfliao}@cqu.edu.cn).

Q. Zhang, Y. Xue and L. Chu are with the Department of Computing and Software, McMaster University, Hamilton, L8S 4L8, Canada (email: {zhangq16, xuey45, chul9}@mcmaster.ca).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes the appendices on additional experimental results of proposed methods. This material is 2.6 MB in size.

gap is critical not only to advance existing attacks but also to establish more rigorous benchmarks for comprehensively evaluating the robustness of DNN-based image watermarking models against adversarial attacks.

In this paper, we fill this gap by proposing a novel attack named **fast and effective overwrite attack (FEOA)**, which can effectively overwrite the watermark messages of many watermarked images in milliseconds without damaging much image quality. In particular, we demonstrated the existence of a single noise image, when directly added to many different watermarked images, can effectively overwrite their true watermark messages. Since attacking a watermarked image only requires adding this noise image to it, and the noise image does not need to be re-trained or fine-tuned to attack new watermarked images, our attack is extremely fast. Moreover, the damage to the image quality of watermarked images can be easily mitigated by reducing the L_2-norm of the noise image. We also extended FEOA to a more effective version named **adaptive FEOA (AFEOA)**, which trains k different noise images and significantly improves attack effectiveness by adaptively selecting the most effective noise image to launch each attack. Fig. 1 illustrates the key ideas of FEOA and AFEOA. Our contributions are listed as follows.

1) We propose a novel task of fast and effective overwrite attack (FEOA) against DNN-based image watermarking models, which aims to swiftly overwrite the watermark messages of many watermarked images by adding the same noise image to them. To validate the existence of such a noise image, we formulate the task of FEOA as a constrained optimization problem and solve it by projected gradient descent to produce a single noise image that is highly effective in overwriting watermark messages.
2) By applying the principle of divide and conquer, we extend FEOA to adaptive FEOA (AFEOA) to significantly improve attack effectiveness. AFEOA trains a set of k different noise images and integrates a *selective neural network* to choose the most effective noise image for each attack. We formulate the task of AFEOA as a continuous optimization problem and efficiently solve it by proposing a new training method to train the k noise images and the selective neural network simultaneously.
3) We conduct extensive experiments to compare the attack performance of FEOA and AFEOA with eight baselines when attacking seven state-of-the-art DNN-based image watermarking models and one classic image watermarking method. Experimental results demonstrate the superiority of our novel attack methods over the baselines in achieving high attack time efficiency and high attack effectiveness. Our source code is at https://github.com/ShaoxinLi/Fast-and-Effective-Overwrite-Attack.

II. Related Works

To the best of our knowledge, how to launch fast and effective overwrite attacks against deep neural network (DNN)-based image watermarking models [1], [2], [7]–[10], [12], [14]–[16], [32]–[34] is a novel task that has not been systematically studied in the literature. It is broadly related to the following existing attack methods.

The **classic attacks**, such as Gaussian noise [8], [17], [27], Gaussian blur [1], [2], [7], [27] and JPEG [9], [27], [32], are often adopted as benchmark methods to attack watermarked images. These methods can launch fast attacks in real time due to their simplicity. However, they do not pose a big threat to modern DNN-based image watermarking models because most of them are designed and empirically demonstrated to be robust to classic attacks [8]–[10], [12].

The **image-dependent attacks** [23], [25], [26], [28], [29] attack a watermarked image by adding a specifically trained noise image to it. Such a noise image is often referred to as a *perturbation* [23], [25], [26], thus making image-dependent attacks also known as *perturbation attacks*. However, since attacking each new watermarked image requires training a new perturbation, and training each perturbation takes several minutes in the worst case by solving a complex optimization problem, these methods are not applicable for launching real-time attacks and tend to be slow when attacking a large amount of watermarked images. For example, WEvade [23] takes over 22 hours to attack 1,000 watermarked images. In comparison, while our attack methods are also perturbation attacks since they also add perturbations (i.e., noise images) to watermarked images, they are extremely fast in attacking new watermarked images. This is achieved by training the perturbations only once in an offline manner and then directly applying them to attack new watermarked images without any fine-tuning or re-training. As shown by our experiments in Section VI-D, FEOA and AFEOA cost less than 20 milliseconds and 3 seconds, respectively, to attack 1,000 watermarked images.

The **image-regeneration attacks** [24]–[27], [30], [31] train a generative model that takes a watermarked image as input and generates an attacked image without a watermark. Once trained, the generative model can be used to launch fast attacks since each attack only needs a simple forward pass of a watermarked image through the generative model. Some image-regeneration attacks [24], [30] adopt a different attack setting from ours, because [24] requires to know the secret watermark messages of the watermarked images to be attacked, and [30] requires access to the training dataset used to train victim watermarking models. As discussed later in Section III, our attack methods are working on a more practical attack setting than [24], [30]. The other image-regeneration attacks [25]–[27], [31] utilize a diffusion model [35], [36] or a variational autoencoder [27], [31] to generate attacked images. However, since these models cannot precisely control the content of output image, they often overly smooth the attacked image [26], [27], [31]. This produces a low peak signal-to-noise ratio (PSNR) between the attacked image and the original image, thus damaging the utility of the attacked image. In summary, the proposed FEOA and AFEOA are novel attack methods that are substantially different from the image-regeneration attacks, because FEOA and AFEOA do not train a generative model to generate attacked images.

In addition, a few recent attacks [37], [38] do not fall in the above categories. [37] is specifically designed to target the watermarking model of Stable Signature [14], and therefore, this attack method cannot be directly applied to other DNN-based image watermarking models. [38] extracts the water-

mark pattern corresponding to a specific watermark message and then uses it to conduct attacks. However, this watermark pattern is only effective against images watermarked with the same message and does not perform well when attacking images watermarked with different messages.

III. Threat Model

In this section, we first give a brief introduction to the general encoder-decoder framework of DNN-based image watermarking models. Then, we present our threat model used in this work.

A DNN-based image watermarking model [1], [2], [7]–[10], [12], [14]–[16], [32]–[34], denoted by S, often involves two components: an encoder and a decoder. Given an **original image** I_o that does not carry a watermark, the **encoder**, denoted by E, is a deep neural network to add a message w into I_o. This produces a **watermarked image**, denoted by I_w. The **message** $w \in \{0,1\}^l$ is a sequence of l bits and it is often called a watermark or a watermark message. We write this process to add a message as $I_w = E(I_o, w)$. The **decoder**, denoted by D, is a deep neural network that extracts the message w from I_w. We write this process to extract a message as $w = D(I_w)$.

Given a victim DNN-based image watermarking model S to be attacked, we consider the following **threat model**, which describes an attacker's access to different types of information.

1) **The decoder** D. We consider the following two versions of the attacker's access to D: *a) White-box version*: the attacker has complete knowledge of D, such as its model architecture and model parameters; and *b) Black-box version*: the attacker has zero knowledge about D and cannot even use D as a black box to decode messages.
2) **The encoder** E. The attacker has zero knowledge about the internal mechanism of E, such as the model architecture, model parameters and watermarking algorithm. However, the attacker can pretend to be a normal user of E and use E as a black box to generate watermarked images with known messages.
3) **The training images**. Denote by X_O the training dataset used to train S. The attacker cannot access the images in X_O; and the attacker has zero knowledge about the distribution of the images in X_O.
4) **The publicly available images**. The attacker can access publicly available images that are not watermarked by S.
5) **The messages of target watermarked images to attack**. For each target watermarked image I_w, the attacker does not know the message w carried by I_w.

IV. Fast and Effective Overwrite Attack

In this section, we introduce and formulate the fast and effective overwrite attack against DNN-based image watermarking models. We first define the task of fast and effective overwrite attack as follows.

Definition 1. *Given a victim DNN-based image watermarking model S and a real-valued threshold $\xi > 0$, the task of **fast and effective overwrite attack (FEOA)** is to train a single noise image, denoted by δ, such that*

1) for each watermarked image I_w produced by S, adding δ to I_w will change the message w carried by I_w to a different message; and
2) the L_2-norm of δ, denoted by $\|\delta\|_2$, is not larger than ξ.

The first condition in Definition 1 requires $D(I_w + \delta) \neq D(I_w)$, which means adding δ to I_w overwrites the message of I_w to a different one. When this happens, we say the watermarked image I_w is **successfully attacked** by the noise image δ. The second condition limits the L_2-norm of δ by ξ. By plugging this constraint into the definition of PSNR, we can derive the lower bound between the watermarked image I_w and the attacked image $I_w + \delta$, that is,

$$\text{PSNR}(I_w, I_w + \delta) \geq 10 \cdot \log_{10}\left(\frac{d \cdot \max^2}{\xi^2}\right), \tag{1}$$

where $\max = 255$ is the maximum possible pixel value of the image and d is the total number of pixels in I_w. Apparently, a smaller value ξ leads to a larger PSNR, thus reducing the damage to the utility (i.e., quality) of the attacked image. Please also refer to Fig. 1(a) for the key idea of the above FEOA task.

To train the single noise image δ, we first obtain a training dataset consisting of N watermarked images with known messages, denoted by X. An attacker can obtain the training images in X by pretending to be a normal user of the watermarking system S. In this way, the attacker can generate a set of watermarked images with known messages by using the encoder of S to watermark publicly available images. Here, X is different from the training dataset X_O used to train S, because the attacker has zero knowledge about X_O.

The task of FEOA aims to train δ based on X. We formulate this task as a **constrained optimization problem**, that is,

$$\begin{aligned} \min_{\delta} \quad & -\frac{1}{|X|}\sum_{I_w \in X} L\big(F(I_w + \delta), w\big), \\ \text{s.t.} \quad & \|\delta\|_2 \leq \xi, \end{aligned} \tag{2}$$

where δ is the noise image to train, ξ is the upper bound of the L_2-norm of δ, $I_w \in X$ is a training image, w is the true message of I_w, and $|X|$ is the size of X, The loss function $L(\cdot,\cdot)$ computes the distance between two l-dimensional vectors, that is,

$$L(w', w) = -\sum_{i=1}^{l} w_i \log w'_i + (1 - w_i)\log(1 - w'_i), \tag{3}$$

where w'_i and w_i are the i-th entries in the vectors w' and w, respectively. When using $L(w', w)$ in (2), we have w to be the true message of I_w, and $w' = F(I_w + \delta)$ is the message decoded by the function $F(\cdot)$ from the attacked image $I_w + \delta$. The i-th entry of w' indicates the likelihood for the i-th bit of the decoded message to be equal to one. In summary, (2) aims to train a single noise image δ, such that the message decoded from $I_w + \delta$ can be as different from the true message w as possible. This fulfills the purpose of overwriting the messages of watermarked images.

The optimization problem in (2) can be easily solved by the projected gradient descent (PGD) [39], which requires to know $F(\cdot)$ in order to pass gradients to δ. Depending on the

attacker's access to the decoder D of S, we can obtain $F(\cdot)$ in the following two ways.

In the **white-box version** where the attacker has access to the decoder D of S, $F(\cdot)$ can be obtained from D by excluding the last layer of D. This is because the second-to-last layer of D produces the likelihood for each bit of the decoded message to be equal to one [23].

In the **black-box version** where D is not accessible, the attacker can train a surrogate model, denoted by F_S, to act as the proxy of $F(\cdot)$. Specifically, F_S is trained by solving the following optimization problem:

$$\min_{\theta} \quad \frac{1}{|X_S|} \sum_{I_w \in X_S} L\big(F_S(I_w), w\big), \tag{4}$$

where θ is the model parameters of F_S, X_S is a training dataset of watermarked images obtained in the same way as how we obtain X, and $|X_S|$ is the size of X_S. A detailed description of X_S is introduced later in Section VI-A. Inspired by prior works [7], [12], we choose a ResNet-50 [40] with l sigmoid-activated output neurons in the last layer as the model architecture of F_S. ResNet-50 is a deep convolutional neural network consisting of 50 layers with residual connections and we use l sigmoid-activated output neurons in the last layer such that $F_S(I_w)$ produces the likelihood that each bit in the decoded message is equal to one. This model architecture has been shown to be effective in extracting watermark messages from watermarked images [7], [12] and we empirically find good attack performance of our methods when adopting it as F_S in the experiments. We also investigate the impact of different model architectures of F_S on the attack performance in Section VI-F. After training F_S, we can replace the $F(\cdot)$ in (2) by F_S to train δ.

Given the trained noise image δ, attacking a new watermarked image I_{new} is as simple as adding δ to I_{new} and clipping the pixel values of $I_{\text{new}} + \delta$ to the range of $[0, 255]$. The noise image δ does not need to be re-trained or fine-tuned to attack different watermarked images. Due to the simplicity of launching an attack, FEOA is able to achieve an extremely fast attack speed, which opens the door to effectively attacking a large number of watermarked images at scale.

The effectiveness of FEOA stems from the inherent vulnerability of DNN-based decoders to input noise. Although DNN-based image watermarking models exhibit robustness against classic attacks, they all extract watermark messages relying on the DNN-based decoder D, which is inherently susceptible to input noise due to its nonlinear nature and high-dimensional input space [41]–[43]. This enables a carefully crafted noise image δ that, when added to input watermarked images, can induce significant deviations in D's output, thereby preventing D from extracting true messages. Furthermore, as revealed in [44], such a noise often has a dominant contribution to the DNN's response. This dominance allows δ to overwrite the true messages of many watermarked images by misleading D into extracting fake messages. These fundamental limitations of DNN-based decoders underpin the effectiveness of FEOA.

Moreover, we find that the noise image trained by FEOA is not unique. Due to the non-convexity of (2), there exist many different noise images, each of which can successfully attack a different set of watermarked images. This motivates us to extend FEOA to a more advanced adaptive version, which trains k noise images to further improve attack effectiveness based on the principle of divide and conquer.

V. Adaptive Fast and Effective Overwrite Attack

In this section, we first introduce how to extend the task of FEOA to the task of adaptive FEOA. Then, we formulate this task as an optimization problem and describe how to solve the problem.

A. Task Definition

As shown in Fig. 1(b), the *key idea* of adaptive FEOA is to train a set of k noise images and adaptively select the most effective noise image to attack each new watermarked image. The selection is made by a **selective neural network** that is trained together with the noise images. In this way, if each of the k noise images successfully attacks a different set of watermarked images, then we will successfully attack the union of the k sets of watermarked images, which leads to higher attack effectiveness. Following this key idea, we extend the task of FEOA in Definition 1 to the following task.

Definition 2. *Given a victim DNN-based image watermarking model S and a real-valued threshold $\xi > 0$, the task of **adaptive fast and effective overwrite attack (AFEOA)** is to train a set of k noise images, denoted by $P = \{\delta_1, \ldots, \delta_k\}$, and a selective neural network, denoted by h_β, such that*

1) *for each watermarked image I_w produced by S, h_β selects the most effective noise image in P to attack I_w, which will change the message w carried by I_w to a different message; and*
2) *for each noise image $\delta \in P$, $\|\delta\|_2 \leq \xi$.*

AFEOA is "adaptive" because the most effective noise images for different watermarked images may be different, and AFEOA uses h_β to adaptively select the most effective noise image when attacking each watermarked image I_w. Only the selected noise image is used to attack I_w. Therefore, when using k noise images to perform AFEOA, the target watermarked images fall into k separate subsets, where h_β selects the same noise image to attack every I_w in the same subset. This implements the principle of divide and conquer, where each noise image is used to perform the FEOA attack in each subset of watermarked images. As a result, AFEOA will improve the attack effectiveness because the set of watermarked images successfully attacked by AFEOA will be the union of the successfully attacked images in each subset.

B. Formulating the AFEOA Task

In this subsection, we first introduce the design details of the selective neural network h_β. Then, we formulate the AFEOA task into a continuous optimization problem.

The selective neural network h_β is a DNN with parameters β. Given a watermarked image I_w as input, $h_\beta(I_w)$ outputs a k-dimensional probability vector, that is, $h_\beta(I_w) = \pi = \{\pi_1, \ldots, \pi_k\}$. Ideally, the i-th entry of π, denoted by π_i,

represents the probability of selecting the i-th noise image $\delta_i \in P$ as the most effective noise image to attack I_w. We adopt a SqueezeNet [45] with k softmax-activated output neurons in the last layer as the model architecture of h_β. SqueezeNet is a lightweight convolutional neural network that consists of 18 layers. It has been shown to perform well in many different vision tasks [46], [47], and we empirically observe good attack performance of AFEOA in the experiments when employing SqueezeNet as h_β. We also investigate the impact of different model architectures of h_β on the attack performance of AFEOA in Section VI-F.

Denote by z a categorical variable following the categorical distribution characterized by π. Since $\pi = h_\beta(I_w)$, the distribution of z is characterized by $h_\beta(I_w)$, thus we write $z \sim h_\beta(I_w)$. We encode z as a k-dimensional one-hot vector, where the i-th entry of z being equal to one means the i-th noise image $\delta_i \in P$ is selected by $h_\beta(I_w)$. The selection process is implemented as $C(P, z) = \delta_1 \cdot z_1 + \ldots + \delta_k \cdot z_k$, where z_i, $i \in \{1, \ldots, k\}$, is the i-th entry of z.

Now, we extend (2) to formulate the task of AFEOA as the following **continuous optimization (CO) problem**.

$$\begin{aligned} \min_{\beta, P} \quad & -\frac{1}{|X|} \sum_{I_w \in X} \mathbb{E}_{z \sim h_\beta(I_w)} \Big[L\big(F(I_w + C(P, z)), w\big) \Big], \\ \text{s.t.} \quad & \|\delta\|_2 \leq \xi, \forall \delta \in P, \end{aligned} \tag{5}$$

where I_w is a watermarked image in the training dataset X, w is the true message of I_w, $F(\cdot)$ and $L(\cdot, \cdot)$ are the same functions as in (2).

Minimizing the objective function in (5) requires to maximize the expectation term $\mathbb{E}_{z \sim h_\beta(I_w)}[\cdot]$ for each watermarked image $I_w \in X$, which can also be written as $\mathbb{E}_{z \sim h_\beta(I_w)}[\cdot] = \sum_{i=1}^{k} \pi_i L(F(I_w + \delta_i), w)$. Denoted by $\delta_{i^*} \in P$ the most effective noise image in attacking I_w, the distance between $F(I_w + \delta_{i^*})$ and w will be the largest among all the noise images in P, which produces the largest value of $L(F(I_w + \delta_i), w)$. Thus, in order to maximize $\sum_{i=1}^{k} \pi_i L(F(I_w + \delta_i), w)$, the probability corresponding to δ_{i^*}, denoted by $\pi_{i^*} \in \pi$, should be the highest among all the probabilities in π. Since $\pi = h_\beta(I_w)$, this means the noise image in P corresponding to the highest probability output by $h_\beta(I_w)$ is the most effective noise image in attacking I_w. Therefore, by minimizing the objective function in (5) to train the set of k noise images in P and the selective neural network h_β, the noise image selected by $h_\beta(I_w)$ with the highest probability is often the most effective noise image in P to attack I_w. This fulfills the purpose of AFEOA. We verify the validity of h_β in selecting the most effective noise image in Appendix F.

C. Solving the CO Problem

Solving the CO problem in (5) needs to explicitly compute the expectation term $\mathbb{E}_{z \sim h_\beta(I_w)}[\cdot]$ for every watermarked image $I_w \in X$. However, this could be computationally expensive when the number of noise images k is large, because it requires computing $L(F(I_w + C(P, z)), w)$ for k times, each for one possible value of z.

To address this issue, we apply the Gumbel softmax trick [48] to convert the optimization problem in (5) into

$$\begin{aligned} \min_{\beta, P} \quad & -\frac{1}{|X|} \sum_{I_w \in X} \mathbb{E}_{\mathbf{g} \sim Gumbel(0,1)} \Big[L\big(F(I_w + C(P, z')), w\big) \Big], \\ \text{s.t.} \quad & \|\delta\|_2 \leq \xi, \forall \delta \in P, \end{aligned} \tag{6}$$

where $\mathbf{g} = [g_1, \ldots, g_k]$ is a k-dimensional vector, the entries in $\mathbf{g}$ are independent random variables following the $Gumbel(0, 1)$ distribution [48], and $z' = [z'_1, \ldots, z'_k]$ is a k-dimensional vector with the i-th entry defined as

$$z'_i = \frac{\exp\big((\log \pi_i + g_i)/\tau\big)}{\sum_{j=1}^{k} \exp\big((\log \pi_j + g_j)/\tau\big)}. \tag{7}$$

Here, the parameter $\tau > 0$ is a temperature parameter used by the Gumbel softmax trick [48]. A smaller temperature τ makes z' closer to a one-hot vector, but it also causes a larger variance of the gradients computed by back-propagation when optimizing the objective function in (6) [48]. Following [49], [50], we start by setting τ to a large value and then gradually anneal it to a smaller value as the training continues. We adopt the following annealing schedule:

$$\tau = \max\big(0.01, \exp(-1e - 4 \cdot t)\big), \tag{8}$$

where t is number of training iterations and τ is updated every 2000 iterations.

By applying the above Gumbel softmax trick, we can estimate the expectation term $\mathbb{E}_{\mathbf{g} \sim Gumbel(0,1)}[\cdot]$ in (6) by sampling the entries of $\mathbf{g}$ from the $Gumbel(0, 1)$ distribution. This is much more efficient than explicitly computing the expectation term $\mathbb{E}_{z \sim h_\beta(I_w)}[\cdot]$ in (5) when k is large.

We solve the optimization problem in (6) to train P and h_β by stochastic gradient descent. Following the routine of Gumbel softmax trick [49], [50], for each gradient step, the value of the objective function is computed on a batch of training images X_B sampled from X and a vector $\mathbf{g}$ sampled from $Gumbel(0, 1)$. Then, the gradients are computed by standard back-propagation, and P and β are updated alternatively. We update the noise images in P by the projected gradient descent (PGD) [39] to ensure each noise image $\delta \in P$ is a feasible solution satisfying the convex constraint $\|\delta\|_2 \leq \xi$ in (6). We update β using the ADAM optimizer [51], which is a classic method to train neural networks. The training algorithm is summarized in Algorithm 1.

Once P and h_β are trained, we can use the noise images in P and h_β to quickly perform adaptive overwrite attacks on new watermarked images in milliseconds. Specifically, to attack a new watermarked image I_{new}, we first compute the probability vector $\pi = h_\beta(I_{\text{new}})$. Then, we select the noise image corresponding to the highest probability in π to attack I_{new}. Denote by δ_{i^*} the selected noise image, we attack I_{new} by simply adding δ_{i^*} to I_{new} and clipping the pixel values of the resulting attacked image $I'_{\text{new}} = I_{\text{new}} + \delta_{i^*}$ to the range of $[0, 255]$. Since selecting the noise image only requires a forward pass of h_β and the noise images in P do not need to be further fine-tuned or re-trained when launching new attacks, performing AFEOA to attack a single watermarked image only

Algorithm 1: Solving the CO problem

Inputs : The decoder D or the surrogate model F_S, the training dataset X, the number of noise images k and the threshold ξ.

Outputs: The trained noise images in P and the trained selective neural network h_β.

Initialize each noise image in P by zeros.
Initialize β of h_β by the Kaiming initialization [52].
Set the number of training iterations $t = 1$.
do
  Sample a batch of training images X_B from X and sample $\mathbf{g}$ from $Gumbel(0, 1)$.
  Use X_B and $\mathbf{g}$ to compute the gradients of P and update P by the projected gradient descent [39].
  Repeat step-5 to sample a new pair of X_B and $\mathbf{g}$.
  Use X_B and $\mathbf{g}$ to compute the gradients of β and update β by the ADAM optimizer [51].
  If $t \bmod 2000 = 0$ then update τ by (8).
  Update $t \leftarrow t + 1$.
while *not converge*;
return P and h_β.

Algorithm 2: Conducting an AFEOA attack

Inputs : The trained noise images in P, the trained selective neural network h_β and a new watermarked image I_{new}.

Output: The attacked image I'_{new}.

Compute: $\pi = h_\beta(I_{\text{new}})$.
Select: $i^* \leftarrow \text{argmax}_i\, \pi_i$.
Attack: $I'_{\text{new}} \leftarrow I_{\text{new}} + \delta_{i^*}$.
Clip the pixel values of I'_{new} to the range of $[0, 255]$.
return I'_{new}.

takes about 2 milliseconds on an NVIDIA RTX 3090 GPU. We summarize the attacking process in Algorithm 2.

The effectiveness of AFEOA not only stems from the inherent vulnerability of DNN-based decoders to input noise, but is also amplified by implementing the principle of divide and conquer. By training k noise images such that each of them focuses on attacking a large but substantially different subset of watermarked images, the subsets of images successfully attacked by different noise images are complementary to each other. Hence, when using the selective neural network h_β to adaptively select the most effective noise image to attack each watermarked image, the set of images successfully attacked by AFEOA will be the union of the successfully attacked images in each of the k subsets. Therefore, compared to using a single noise image to attack all watermarked images, AFEOA can successfully attack a larger set of watermarked images, thus offering high attack effectiveness.

VI. Experiments

In this section, we systematically evaluate the performance of the proposed FEOA and AFEOA against DNN-based image watermarking models. We first present the experimental settings and our evaluation method. Then, we compare the performance of FEOA and AFEOA with baselines. Next, we analyze the effect of the number of noise images k on the performance of AFEOA. We also conduct an ablation study to investigate the impact of different design choices on the performance of our methods. Last, we discuss potential limitations of our methods.

Due to the limit of space, we present the following experiments in the Appendix of the *supplementary material*. We empirically verify the lower bound of PSNR given by (1) in Appendix E and the validity of the selective neural network h_β in selecting the most effective noise image in Appendix F. We also conduct a case study in Appendix G to show the watermarked images attacked by different attack methods.

A. Experimental Settings

Baselines. We compare the proposed attack methods with eight baselines, including 1) five advanced attack methods, such as WMAttacker [27], Diffpure [26], RinseDiff [31], RinseVAE [31] and steganalysis-based removal (SBR) [38]; and 2) three classic attack methods, such as Gaussian noise [8], [17], [27], Gaussian blur [7], [27] and JPEG [9], [27], [32].

Our methods. For each of FEOA and AFEOA, we implement both the white-box version and the black-box version of it. The white-box versions of FEOA and AFEOA are denoted as FEOA-WB and AFEOA-WB, respectively; and their black-box counterparts are denoted as FEOA-BB and AFEOA-BB, respectively.

Victim models. We evaluate the performance of different attack methods against seven representative DNN-based image watermarking models, including HiDDeN (HD) [1], UDH [2], SSL [7], RoSteALS (ROS) [12], Stable Signature (SS) [14], TreeRing (TR) [15] and CRIW [16]. In addition, we also attack the traditional image watermarking method DwtDctSvd (DDS) [53]. The decoder of DDS is not differentiable, thus we cannot employ FEOA-WB and AFEOA-WB to attack DDS. We only attack DDS by FEOA-BB and AFEOA-BB, as the surrogate model F_S of DDS is differentiable.

Datasets. We use three benchmark datasets for evaluation, including COCO [54], ImageNet [55] and Conceptual Caption [56]. The usage of the datasets is explained as follows.

1) *We use the training datasets of COCO and ImageNet to train victim models.* For HD, UDH, ROS and CRIW, we train their watermarking models on 10,000 images sampled from the training datasets of COCO and ImageNet, respectively. The rest of the watermarking models do not use the datasets for training, because SSL and DDS do not need to train a watermarking model, and we use the pre-trained models of SS and TR released in their public repositories.
2) *We use the testing and validation datasets of COCO and ImageNet to generate the testing dataset of watermarked images, which are used to evaluate the performance of different attack methods.* For each of HD, UDH, ROS, CRIW, SSL and DDS, we generate 1,000 watermarked images from 1,000 original images sampled from the testing dataset of COCO and the validation dataset of ImageNet,

respectively. Since SS and TR use text prompts to generate watermarked images, we follow [14] to use the captions of 5,000 images sampled from the validation dataset of COCO as the prompts to generate 5,000 watermarked images.

3) *We use the training dataset of Conceptual Caption to extract the watermark pattern of SBR and also train the noise images, the selective neural network h_β and the surrogate model F_S of our methods.* We first sample 10,000 original images from the training dataset of Conceptual Caption. Then, for each watermarking model, we use the original images to generate 10,000 watermarked images with the same message, which are used to extract the watermark pattern of SBR. We also use the same set of original images to generate 10,000 watermarked images with uniformly sampled messages for each watermarking model. These images are used as the training dataset X in (2) and (6) to train the noise images and h_β of our methods. To train F_S for FEOA-BB and AFEOA-BB, we independently repeat the above steps to generate 10,000 new watermarked images, which are used as the training dataset X_S in (4) to train F_S.

Implementation details. Here, we introduce the implementation details of the victim models, the baseline methods and our methods.

All the victim models are implemented using their public source codes and default configurations [1], [2], [7], [12], [14]–[16], [53]. For the compared baseline methods, we use their publicly available source codes [23], [26], [27], [31], [38] and default optimal parameters.

For FEOA-WB and FEOA-BB, we set the learning rate of the PGD to 10^{-3}, the batch size to 16 and use 50 training epochs. For AFEOA-WB and AFEOA-BB, we set the learning rates of the ADAM optimizer and the PGD to 10^{-3}, using $k = 150$ and 100 training epochs if not otherwise specified. To train the surrogate model F_S, we use the ADAM optimizer and set the learning rate to 10^{-3}, batch size to 16 and the training epochs to 100. In all experiments, we set l to the default message length of the victim model. All the experiments are conducted on a server with an NVIDIA RTX 3090 GPU, 64GB main memory and an Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz.

B. Evaluation Method

We evaluate the attack performance of an attack method by its attack effectiveness and the quality of attacked images. We also evaluate the attack efficiency of each attack method.

Attack effectiveness. We evaluate the effectiveness of an attack method in overwriting watermark messages in the context of a typical image watermarking task named *image attribution* [14], [17]. The goal of image attribution is to identify the true message w carried by a watermarked image I_w from a set of candidate messages, denoted by $W = \{w_1, \ldots, w_T\}$. Each watermarked image I_w may be classified by the decoder of an image watermarking model as either "not watermarked" if the decoder cannot decode a message from I_w, or as watermarked with a specific message $w_i \in W$ if w_i is successfully decoded from I_w. Following [14], [17], we set the anticipated false positive rate of the image watermarking model S to 10^{-4}, and then evaluate the attribution accuracy (ACC) [14] of S by the percentage of the watermarked images whose true messages are correctly identified by the decoder of S. At last, we evaluate the effectiveness of an attack method by the *attribution accuracy drop*, that is,

$$\Delta_{\text{ACC}} = \text{ACC}_{\text{before}} - \text{ACC}_{\text{after}}, \tag{9}$$

where $\text{ACC}_{\text{before}}$ and $\text{ACC}_{\text{after}}$ are the attribution accuracies on the testing dataset before and after the watermarked images are attacked. Each $\frac{1}{T}$ proportion of the watermarked images in the testing dataset are watermarked by one of the T messages in $W = \{w_1, \ldots, w_T\}$, which are uniformly sampled from the space of $\{0,1\}^l$. A larger value of Δ_{ACC} implies higher attack effectiveness. In our experiments, we set $T = 1$ when attacking SS, as the pre-trained models released in its public repository only support embedding a single message. For the other victim models, we use a default value of $T = 10$.

The quality of attacked images. Following the setting of WMAttacker [27], we evaluate the quality of an attacked image by measuring the peak signal-to-noise ratio (PSNR) between the attacked image and its corresponding original image that is not watermarked by any image watermarking model. We report the *average PSNR* of all the images in the testing dataset. In addition, we also evaluate the *Fréchet Inception Distance (FID)* [57] between the attacked images and their corresponding original images. A larger average PSNR and a smaller FID mean higher quality of attacked images, thus implying the attack method is causing less damage to the utility of attacked images. An exceptional case when evaluating the quality of attacked images is TR [15]. Since TR significantly modifies the content of an original image to produce a watermarked image, the content of an attacked image is similar to the watermarked image but quite different from the original image (see an example in the second column of Fig. 10(a) in the Appendix). As a result, the average PSNR and FID between the attacked images and the original images are not meaningful. Therefore, for TR, we report the average PSNR and FID between the watermarked images and the attacked images.

Attack efficiency. We evaluate the attack efficiency of each attack method by *total attack time*, which is the overall time cost to generate the attacked images for the 1,000 watermarked images produced by HD, UDH, ROS, SSL, CRIW and DDS, respectively, and for the 5,000 watermarked images produced by SS and TR, respectively. Since the training of WMAttacker, Diffpure, RinseDiff, RinseVAE, SBR and our methods are done in an offline manner, we do not count the offline training time cost as part of the total attack time. A smaller total attack time indicates higher attack efficiency. We report the total attack time in milliseconds by default.

C. Attack Performance

In this subsection, we compare the attack performance of all the attack methods when evaluating the quality of attacked images by average PSNR; and the attack performance when evaluating the quality of attacked images by FID is reported

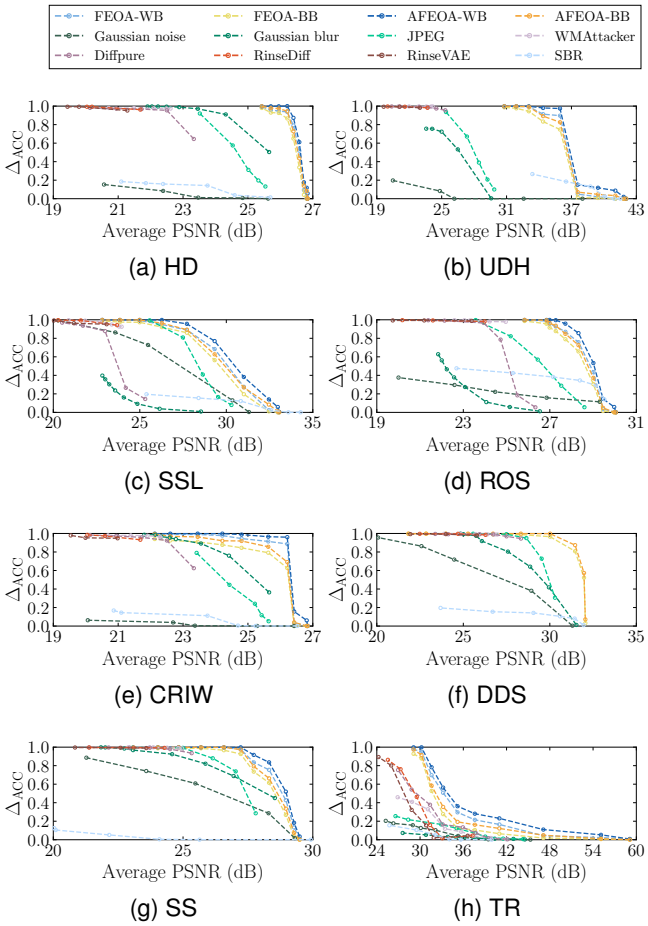


Fig. 2. The Δ_{ACC} and average PSNR on COCO. The caption of each subfigure shows the name of victim model.

in Appendix B. Fig. 2 shows the results of Δ_{ACC} and average PSNR on COCO. Since the results on ImageNet resemble those on COCO, we provide them in Fig. 6 of Appendix B and focus on analyzing the results in Fig. 2 for the rest of this subsection.

For each attack method, we report its Δ_{ACC} at different levels of average PSNR. The average PSNR of each attack method is controlled by tuning the amount of noise added to the attacked images. We introduce the details of how to tune the amount of noise for each attack method in Appendix A. In each subfigure of Fig. 2, a curve that is closer to the upper right corner indicates that the corresponding attack method achieves higher attack effectiveness (i.e., a higher Δ_{ACC}) while maintaining better image quality (i.e., a higher average PSNR), thereby demonstrating superior attack performance.

As shown in Fig. 2, the curves of our methods, such as FEOA-WB, FEOA-BB, AFEOA-WB and AFEOA-BB, are much closer to the upper right corner than the curves of the other baseline methods. This demonstrates their superior attack performance. We can see that FEOA-WB outperforms FEOA-BB and AFEOA-WB outperforms AFEOA-BB. This is because the white-box versions of FEOA and AFEOA can directly access the decoder of the victim model to train the noise images, but the black-box versions can only access the surrogate model. We can also see that the Δ_{ACC} of AFEOA-WB and AFEOA-BB is higher than that of FEOA-WB and FEOA-BB, respectively, for almost every value of the average PSNR. This indicates the efficacy of AFEOA in further improving attack effectiveness.

The image-regeneration attack methods, such as WMAttacker, Diffpure, RinseDiff and RinseVAE, demonstrate inferior attack performance due to their low average PSNR. These methods cannot achieve a high average PSNR because they either use diffusion models [35], [36], [58] or a variational autoencoder (VAE) [59] to regenerate attacked images, but they cannot control the *intrinsic noise* added by diffusion models or VAE to the attacked images. Here, the intrinsic noise is not the Gaussian noise removed by the denoising step of diffusion models or the noise added by VAE during the encoding stage, it is the absolute difference of pixel values between the attacked image and the target watermarked image. Since the diffusion models and VAE often over-smooth the attacked images [27], they introduce a large amount of intrinsic noise that cannot be reduced by tuning the hyperparameters of WMAttacker, Diffpure, RinseDiff and RinseVAE.

The classic attack methods, such as Gaussian noise, Gaussian blur and JPEG, fail to achieve a large Δ_{ACC} without significantly reducing the average PSNR. The reason is that DNN-based image watermarking models are generally robust to the classic attacks [1], [2], [8], [32], thus a classic attack method has to add a large amount of noise in order to achieve a successful attack. We can also see in Fig. 2(f) that the classic attack methods are relatively more effective in attacking the traditional image watermarking method DDS. However, their attack performance is still inferior to that of our methods.

SBR cannot achieve good attack performance, because the watermark pattern extracted by SBR based on one watermark message does not generalize well in attacking the other target watermarked images with different messages [38].

An interesting finding is the high robustness of TR to many baseline methods. We can see in Fig. 2(h) that many baseline methods cannot achieve a high Δ_{ACC} even with a small average PSNR. This is because TR embeds a message during the reverse diffusion process of diffusion models and extracts the message by inverting this process [15]. In this way, a watermarked image produced by TR has a significant pixel-level difference from its original image, which allows TR to embed a stronger watermark [15], [27]. However, our methods still attain the best attack performance on TR. This is because the inversion process of TR to extract messages is executed using the diffusion model's noise predictor, which is inherently a DNN. Therefore, the watermark extraction mechanism of TR effectively functions as a DNN-based decoder, thus inheriting the vulnerability of DNN-based decoders to input noise. Our methods can exploit this vulnerability to craft noise images that disrupt the inversion process and thus prevent the accurate recovery of embedded messages.

Recall that the noise images and the surrogate model F_S of our methods are trained on a dataset with a distribution significantly different from that of the victim models' training datasets. This raises an interesting question: *how can our methods achieve strong attack performance despite this large distribution gap?* Our answers are as follows.

First, the watermark patterns embedded in watermarked images are largely determined by the encoder E of the victim model, thus the distribution of watermark patterns does not change much for images with a large distribution gap. Second, the surrogate model F_S is trained to recognize the watermark patterns instead of the image contents. Therefore, the performance of F_S in correctly decoding watermark patterns is not affected much by the large distribution gap of training images. Third, the noise images are trained to overwrite the watermark patterns, thus their attack effectiveness is more related to the distribution of the watermark patterns instead of image contents. In summary, as demonstrated by the results in Fig. 2, our methods achieve strong attack performance despite the large distribution gap of training images. This improves the practicality of the proposed attack methods, because an attacker can launch strong attacks by simply using publicly available images for training.

D. Attack Efficiency

In this subsection, we investigate the attack efficiency of different attack methods. Table I reports the total attack time on COCO, where the amount of noise for each attack method is set to achieve the largest Δ_{ACC} in Fig. 2. The results on ImageNet are similar, thus we report them in Table VIII of Appendix C and focus on discussing the results in Table I in the following.

We can see that the smallest total attack time is consistently achieved by FEOA-WB, FEOA-BB and SBR, which take 13$\sim$18 milliseconds to attack the 1,000 watermarked images of HD, UDH, SSL, ROS, CRIW and DDS, and about 85$\sim$92 milliseconds to attack the 5,000 watermarked images of SS and TR. For FEOA-WB and FEOA-BB, such a small total attack time is achieved by their simple attack operation, which generates an attacked image by simply adding the noise image to the watermarked image. Once trained, the noise image does not need any re-training or fine-tuning when attacking new watermarked images, thus enabling us to launch very fast attacks. SBR achieves a comparable total attack time to FEOA-WB and FEOA-BB because it attacks a watermarked image by simply subtracting the watermark pattern from the watermarked image [38]. However, as demonstrated in Section VI-C, SBR cannot achieve high attack effectiveness because the watermark pattern extracted based on one watermark message does not generalize well in attacking the other target watermarked images with different messages [38].

The classic attack methods, such as Gaussian noise, Gaussian blue and JPEG, achieve a small total attack time due to the simplicity of their attack scheme. However, they cost more total attack time than FEOA-WB, FEOA-BB and SBR due to the following reasons: 1) Gaussian noise requires to sample a noise in each attack, which costs more time. 2) Gaussian blur first samples a Gaussian noise and then it does a convolutional operation, thus it costs even more time than Gaussian noise. 3) JPEG compresses images through complex operations, which costs much more time than the previous methods.

Compared to FEOA-WB and FEOA-BB, AFEOA-WB and AFEOA-BB take extra time to choose the most effective noise image by forward passing a watermarked image through the selective neural network h_β. However, considering the significant improvement of attack effectiveness achieved by AFEOA-WB and AFEOA-BB in Section VI-C, the extra time cost is a cost-effective tradeoff.

TABLE I
THE TOTAL ATTACK TIME (MILLISECONDS) ON COCO.

# attacked images	1,000						5,000	
Victim models	HD	UDH	SSL	ROS	CRIW	DDS	SS	TR
FEOA-WB	**14**	**14**	**14**	**17**	**14**	n/a	**85**	**90**
FEOA-BB	**14**	**14**	**15**	**18**	**14**	**14**	**87**	**92**
AFEOA-WB	2,301	2,363	2,276	2,621	2,339	n/a	11,705	11,712
AFEOA-BB	2,342	2,360	2,280	2,616	2,327	2,345	11,890	11,727
Gaussian noise	112	119	114	135	116	118	660	670
Gaussian blur	216	218	211	246	217	225	1,235	1,225
JPEG	12,544	12,531	12,558	13,249	12,582	12,870	67,970	67,960
WMAttacker	50,462	51,295	50,735	58,493	50,569	51,314	292,335	293,170
Diffpure	84,259	84,960	85,603	97,149	85,569	86,324	487,660	486,510
RinseDiff	143,492	152,497	148,683	160,285	147,020	150,502	762,534	772,320
RinseVAE	75,691	76,122	75,349	84,672	74,730	75,606	405,682	410,062
SBR	**14**	**15**	**13**	**18**	**15**	**14**	**87**	**92**

TABLE II
THE TRAINING TIME (MINUTES) OF OUR METHODS.

Dataset	Attack	HD	UDH	SSL	ROS	CRIW	DDS	SS	TR
COCO	FEOA-WB	13	15	12	25	23	n/a	24	26
	FEOA-BB	18	19	18	24	18	18	25	25
	AFEOA-WB	77	81	79	89	97	n/a	83	85
	AFEOA-BB	83	82	80	87	85	80	89	87
ImageNet	FEOA-WB	12	16	13	25	23	n/a	n/a	n/a
	FEOA-BB	18	19	18	24	19	18	n/a	n/a
	AFEOA-WB	76	83	80	88	95	n/a	n/a	n/a
	AFEOA-BB	83	84	79	89	86	78	n/a	n/a

The image regeneration attack methods, such as WMAttacker, Diffpure, RinseDiff and RinseVAE, cost a significantly larger total attack time than the other methods. This is because regenerating images by a diffusion model or VAE costs a lot of time and the image-regeneration process has to be performed in an online manner for each new target watermarked image to attack.

In summary, the key reason for the small total attack time of our methods is that the noise images, the selective neural network h_β and the surrogate model F_S are only trained once in an offline manner; and they do not need to be further re-trained or fine-tuned when attacking new target watermarked images. The offline training of our methods is also efficient. As reported in Tables II and III, on a single NVIDIA RTX 3090 GPU, training the noise images and h_β costs at most 97 minutes; and training F_S costs at most 143 minutes.

E. The Effect of k

In this subsection, we analyze the impact of the number of noise images k on the performance of AFEOA. Since the quality of the watermarked images attacked by AFEOA is mainly determined by the threshold ξ and the attack efficiency of AFEOA is not affected much by k, we focus on investigating how the Δ_{ACC} of AFEOA is affected by k. We use $\xi = 800$ when attacking HD, UDH and DDS and use $\xi = 2000$ when attacking SSL, ROS, CRIW, SS and TR. We adopt $k \in \{1, 50, 100, 150, 200\}$ and the other experimental settings are the same as in Section VI-C. Fig. 3 shows the results of Δ_{ACC} on COCO, where the Δ_{ACC} of FEOA-WB and FEOA-BB is shown by horizontal lines and used as reference lines

TABLE III
THE TRAINING TIME (MINUTES) OF F_S.

Dataset	HD	UDH	SSL	ROS	CRIW	DDS	SS	TR
COCO	119	116	122	138	119	118	140	143
ImageNet	122	117	122	140	117	118	n/a	n/a

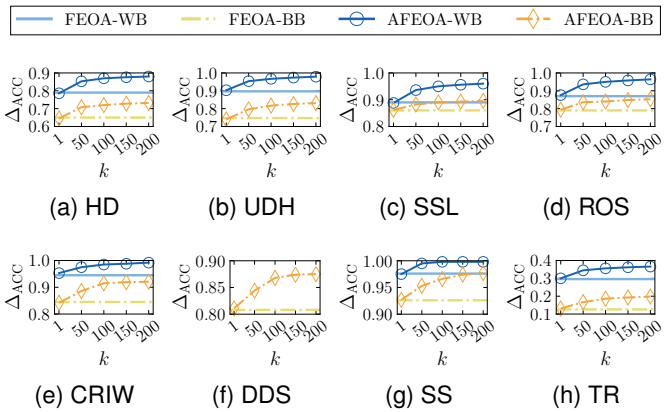


Fig. 3. The Δ_{ACC} for $k \in \{1, 50, 100, 150, 200\}$ on COCO.

to compare with the Δ_{ACC} of AFEOA-WB and AFEOA-BB at different values of k. The results on ImageNet are similar and we discuss them in Appendix D.

As shown in Fig. 3, when $k = 1$, the Δ_{ACC} of AFEOA-WB and AFEOA-BB is close to that of FEOA-WB and FEOA-BB, respectively. This is because $k = 1$ means both AFEOA-WB and AFEOA-BB use a single noise image, thus they degenerate to FEOA-WB and FEOA-BB, respectively. When $1 \leq k < 150$, a larger k improves the Δ_{ACC} of AFEOA-WB and AFEOA-BB. This is because each new noise image may successfully attack some new watermarked images that cannot be successfully attacked by the other noise images, thus using more noise images (i.e., increasing k) increases the number of successfully attacked images. When $k \geq 150$, increasing k does not improve Δ_{ACC} for AFEOA-WB and AFEOA-BB very much. This is due to diminishing marginal utility, that is, the noise images when $k = 150$ have already successfully attacked a large set of watermarked images, which makes it likely that any additional noise images will redundantly target the same images. In summary, we can conclude from Fig. 3 that the attack effectiveness of AFEOA-WB and AFEOA-BB is significantly better than that of FEOA-WB and FEOA-BB, respectively, when $k \geq 150$. This well demonstrates the efficacy of AFEOA in improving attack effectiveness.

F. Ablation Study

In this subsection, we conduct an ablation study to examine the impact of three design choices on the performance of our methods, which include 1) initializing noise images by zeros, 2) using ResNet-50 as the surrogate model and 3) using SqueezeNet as the selective neural network. In the ablation study, each of the three design choices is replaced by several alternatives. We follow the experimental settings in Section VI-E, reporting and discussing the results of the ablation study as follows.

TABLE IV
THE Δ_{ACC} WHEN INITIALIZING NOISE IMAGES BY DIFFERENT METHODS.

Dataset	Attack	Init	HD	UDH	SSL	ROS	CRIW	DDS	SS	TR
COCO	FEOA-WB	Zeros	0.79	0.89	0.89	0.87	0.94	n/a	0.97	0.29
		Gaussian	0.78	0.89	0.90	0.87	0.93	n/a	0.97	0.30
		Uniform	0.79	0.88	0.89	0.88	0.94	n/a	0.86	0.29
	FEOA-BB	Zeros	0.65	0.74	0.86	0.79	0.84	0.99	0.92	0.12
		Gaussian	0.66	0.73	0.85	0.78	0.84	0.99	0.91	0.12
		Uniform	0.65	0.74	0.85	0.80	0.84	0.99	0.91	0.12
	AFEOA-WB	Zeros	0.87	0.97	0.95	0.96	0.98	n/a	0.99	0.36
		Gaussian	0.86	0.95	0.96	0.94	0.96	n/a	0.99	0.33
		Uniform	0.84	0.96	0.93	0.94	0.95	n/a	0.99	0.35
	AFEOA-BB	Zeros	0.73	0.83	0.89	0.84	0.91	0.99	0.97	0.19
		Gaussian	0.70	0.83	0.88	0.82	0.92	0.99	0.95	0.18
		Uniform	0.72	0.80	0.89	0.84	0.91	0.99	0.95	0.19
ImageNet	FEOA-WB	Zeros	0.85	0.73	0.93	0.89	0.96	n/a	n/a	n/a
		Gaussian	0.85	0.73	0.92	0.88	0.95	n/a	n/a	n/a
		Uniform	0.84	0.74	0.93	0.90	0.96	n/a	n/a	n/a
	FEOA-BB	Zeros	0.76	0.55	0.86	0.72	0.82	0.99	n/a	n/a
		Gaussian	0.75	0.54	0.86	0.72	0.82	0.99	n/a	n/a
		Uniform	0.77	0.55	0.86	0.71	0.82	0.99	n/a	n/a
	AFEOA-WB	Zeros	0.94	0.82	0.99	0.96	0.99	n/a	n/a	n/a
		Gaussian	0.92	0.80	0.99	0.95	0.99	n/a	n/a	n/a
		Uniform	0.93	0.79	0.99	0.93	0.98	n/a	n/a	n/a
	AFEOA-BB	Zeros	0.85	0.62	0.92	0.79	0.90	0.99	n/a	n/a
		Gaussian	0.84	0.60	0.91	0.76	0.87	0.99	n/a	n/a
		Uniform	0.86	0.62	0.89	0.77	0.88	0.99	n/a	n/a

TABLE V
THE Δ_{ACC} WHEN USING DIFFERENT MODEL ARCHITECTURES FOR F_S.

Dataset	Attack	F_S	HD	UDH	SSL	ROS	CRIW	DDS	SS	TR
COCO	FEOA-BB	ResNet-50	0.65	0.74	0.86	0.79	0.84	0.99	0.92	0.12
		VGG-16	0.64	0.72	0.86	0.77	0.83	0.99	0.89	0.11
		Inception-V3	0.62	0.73	0.84	0.75	0.82	0.98	0.91	0.11
		SqueezeNet	0.63	0.72	0.85	0.75	0.84	0.98	0.90	0.09
	AFEOA-BB	ResNet-50	0.73	0.83	0.89	0.84	0.91	0.99	0.97	0.19
		VGG-16	0.71	0.81	0.86	0.83	0.91	0.99	0.95	0.17
		Inception-V3	0.74	0.80	0.87	0.85	0.89	0.98	0.97	0.17
		SqueezeNet	0.70	0.79	0.85	0.82	0.87	0.98	0.93	0.15
ImageNet	FEOA-BB	ResNet-50	0.76	0.55	0.86	0.72	0.82	0.99	n/a	n/a
		VGG-16	0.74	0.53	0.85	0.70	0.80	0.99	n/a	n/a
		Inception-V3	0.74	0.53	0.84	0.68	0.81	0.99	n/a	n/a
		SqueezeNet	0.72	0.52	0.84	0.70	0.80	0.98	n/a	n/a
	AFEOA-BB	ResNet-50	0.85	0.62	0.92	0.79	0.90	0.99	n/a	n/a
		VGG-16	0.83	0.60	0.91	0.79	0.88	0.98	n/a	n/a
		Inception-V3	0.84	0.62	0.87	0.77	0.88	0.98	n/a	n/a
		SqueezeNet	0.83	0.60	0.91	0.76	0.87	0.97	n/a	n/a

Table IV reports the results of Δ_{ACC} when noise images are initialized by the zeros initialization, the standard Gaussian initialization and the standard uniform initialization, respectively. We cannot report the results of attacking SS and TR on ImageNet since ImageNet does not provide image captions for SS and TR to generate watermarked images; and we cannot report the results of attacking DDS by the white-box versions of our methods because the decoder of DDS is not differentiable. In addition, since the quality of attacked images and the attack efficiency of our methods are not affected much by different initialization methods, we skip reporting them to save redundancy. We can see from Table IV that using different initialization methods does not have a significant effect on the Δ_{ACC}. Thus, we choose the zeros initialization by default. This also demonstrates that our methods are insensitive to the initialization of noise images and can consistently achieve excellent attack performance.

Table V reports the results of Δ_{ACC} when using ResNet-50 [40], VGG-16 [60], Inception-V3 [61] and SqueezeNet [45] as the model architectures of the surrogate model F_S, respectively. We do not report the results of FEOA-WB and AFEOA-WB since they work in the white-box setting and do not need to train F_S. Again, we cannot report the results of attacking

TABLE VI
THE Δ_{ACC} WHEN USING DIFFERENT MODEL ARCHITECTURES FOR h_β.

Dataset	Attack	h_β	HD	UDH	SSL	ROS	CRIW	DDS	SS	TR
COCO	AFEOA-WB	SqueezeNet	0.87	0.97	0.95	0.96	0.98	n/a	0.99	0.36
		ResNet-50	0.86	0.95	0.94	0.96	0.97	n/a	0.99	0.34
		VGG-16	0.83	0.92	0.90	0.92	0.93	n/a	0.96	0.29
		Inception-V3	0.85	0.94	0.92	0.94	0.96	n/a	0.98	0.32
	AFEOA-BB	SqueezeNet	0.73	0.83	0.89	0.84	0.91	0.99	0.97	0.19
		ResNet-50	0.71	0.83	0.86	0.80	0.90	0.98	0.95	0.17
		VGG-16	0.67	0.79	0.81	0.78	0.87	0.95	0.91	0.14
		Inception-V3	0.69	0.81	0.84	0.80	0.88	0.98	0.93	0.15
ImageNet	AFEOA-WB	SqueezeNet	0.94	0.82	0.99	0.96	0.99	n/a	n/a	n/a
		ResNet-50	0.93	0.82	0.99	0.97	0.99	n/a	n/a	n/a
		VGG-16	0.89	0.79	0.95	0.91	0.97	n/a	n/a	n/a
		Inception-V3	0.91	0.81	0.97	0.95	0.99	n/a	n/a	n/a
	AFEOA-BB	SqueezeNet	0.85	0.62	0.92	0.79	0.90	0.99	n/a	n/a
		ResNet-50	0.84	0.59	0.90	0.77	0.87	0.99	n/a	n/a
		VGG-16	0.78	0.57	0.86	0.74	0.84	0.96	n/a	n/a
		Inception-V3	0.81	0.59	0.89	0.76	0.87	0.98	n/a	n/a

TABLE VII
THE TOTAL ATTACK TIME (MILLISECONDS) WHEN USING DIFFERENT MODEL ARCHITECTURES FOR h_β.

Dataset	Attack	h_β	HD	UDH	SSL	ROS	CRIW	DDS	SS	TR
COCO	AFEOA-WB	SqueezeNet	2,301	2,363	2,276	2,621	2,339	n/a	11,705	11,712
		ResNet-50	9,476	9,631	9,490	11,305	9,563	n/a	56,732	56,450
		VGG-16	36,186	37,942	36,266	40,078	37,004	n/a	168,592	170,206
		Inception-V3	13,293	13,604	14,068	16,937	13,660	n/a	66,146	67,231
	AFEOA-BB	SqueezeNet	2,342	2,360	2,280	2,616	2,327	2,345	11,890	11,727
		ResNet-50	9,519	9,652	9,629	11,252	9,528	9,403	57,025	56,855
		VGG-16	37,390	36,014	37,686	40,863	36,852	37,520	171,143	172,5114
		Inception-V3	14,296	14,842	14,930	17,385	13,760	13,425	67,845	67,307
ImageNet	AFEOA-WB	SqueezeNet	2,424	2,360	2,334	2,868	2,482	n/a	n/a	n/a
		ResNet-50	9,425	9,639	9,520	11,740	9,735	n/a	n/a	n/a
		VGG-16	35,580	36,239	36,288	41,214	37,139	n/a	n/a	n/a
		Inception-V3	14,765	13,882	14,073	16,862	14,463	n/a	n/a	n/a
	AFEOA-BB	SqueezeNet	2,569	2,364	2,348	2,832	2,496	2,502	n/a	n/a
		ResNet-50	9,650	9,823	9,742	12,059	9,733	9,584	n/a	n/a
		VGG-16	37,820	36,926	37,118	41,667	37,147	37,280	n/a	n/a
		Inception-V3	13,790	14,663	14,581	17,272	13,747	13,580	n/a	n/a

SS and TR on ImageNet and we skip reporting the quality of attacked images and the attack efficiency of our methods since they are not affected much by different F_S. We can observe from Table V that while using different model architectures for F_S does not have a large impact on the Δ_{ACC}, adopting ResNet-50 as F_S generally results in a slightly higher Δ_{ACC}. Hence, we choose ResNet-50 as the default model architecture of F_S. These results also indicate that FEOA-BB and AFEOA-BB are not very sensitive to the choices of F_S, which allows them to generally attain excellent attack performance when the decoder of a victim model is not accessible.

Tables VI and VII report the results of Δ_{ACC} and total attack time, respectively, when using SqueezeNet, ResNet-50, VGG-16 and Inception-V3 as the model architectures of the selective neural network h_β. We do not report the results of FEOA-WB and FEOA-BB since they only train a single noise image and do not use h_β. Due to the same reasons as in Table IV, we cannot report the results of attacking SS and TR on ImageNet and the results of attacking DDS by the white-box versions of our methods. We skip reporting the quality of attacked images since it is not affected much by different h_β. As shown in Tables VI and VII, using different choices for h_β does not cause a large influence on the Δ_{ACC}. However, choosing SqueezeNet as h_β leads to a slightly higher Δ_{ACC} in most cases and consistently results in a lower total attack time due to its lightweight network structure. Therefore, we use SqueezeNet as the default model architecture of h_β.

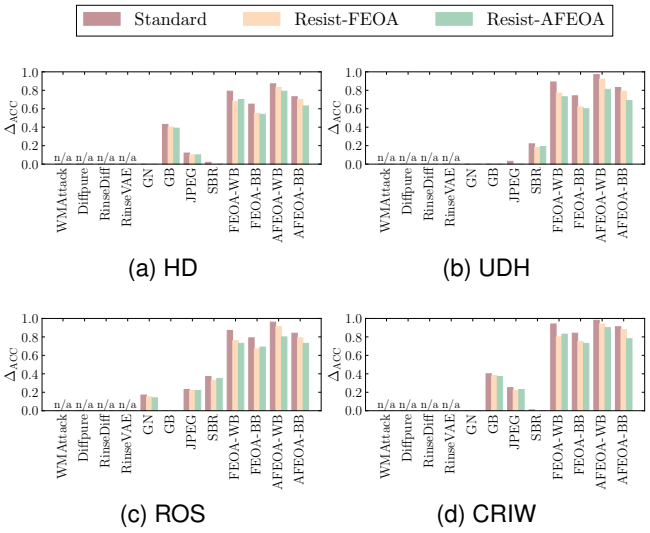


Fig. 4. The Δ_{ACC} on COCO when the victim models are trained in the standard way, resisting FEOA-WB way, and resisting AFEOA-WB way, respectively.

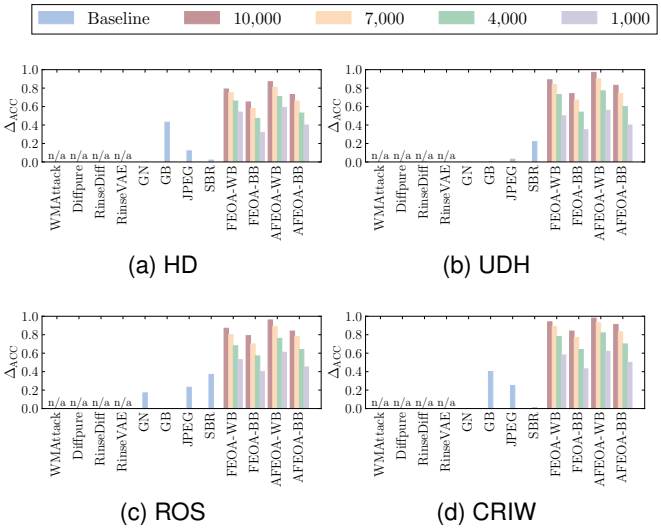


Fig. 5. The Δ_{ACC} on COCO when using 10,000, 7,000, 4,000 and 1,000 watermarked images, respectively, for the training of our methods.

G. Limitations of Our Methods

In this subsection, we discuss potential limitations of our methods. In particular, we focus on two scenarios in which our attacks may fail, including 1) victim models are trained to resist our attacks and 2) the number of watermarked images used by our methods for training the noise images, the selective neural network and the surrogate model is limited. We discuss these two scenarios separately in the following.

In the first scenario, we assume that the owners of the victim models are informed of our attack methods, and thus they can train their models to defend against our attacks. This can be achieved in the same way as how the DNN-based image watermarking models are trained to resist classic attacks [1], [2], [12], [16]. Specifically, given a trained victim model S, the model owner can apply our methods such as FEOA-WB and AFEOA-WB to attack S to generate the noise images. Here we use the white-box versions of our methods because the

decoder D of S is accessible to the owner himself. Then, the owner can continue to train S, during which each watermarked image input to D is attacked by our methods via being added with the noise image. In this way, D can be trained to extract the true messages from the watermarked images attacked by our methods, thus enhancing the ability to resist our attacks.

Fig. 4 shows the results of Δ_{ACC} on COCO when the victim models HD, UDH, ROS and CRIW are trained to resist FEOA-WB and AFEOA-WB, respectively. For comparison, we also show the results of the baseline methods and include the results when these victim models are trained in the standard way, i.e., they are not trained to resist our attacks. For our methods, we use $\xi = 800$ when attacking HD, UDH and $\xi = 2000$ when attacking ROS and CRIW. We tune the amounts of noise for Gaussian noise (GN), Gaussian blur (GB), JPEG and SBR to achieve an average PSNR close to that of our methods. We cannot report the results of WMAttacker, Diffpure, RinseDiff and RinseVAE since they cause too much damage to the image quality and the maximum average PSNR they can achieve is much lower than that achieved by our methods.

As shown in Fig. 4, by training the victim models to resist our attacks, the Δ_{ACC} of our methods is reduced, which demonstrates the effectiveness of the robust training. However, our methods still outperform the compared baseline methods by achieving a higher Δ_{ACC}. This suggests that although our attacks can be mitigated to some extent by the robust training, they still remain very powerful, thus highlighting the necessity to study more effective defenses in future research.

In the second scenario, we assume that the number of watermarked images used by our methods to train the noise images, the selective neural network and the surrogate model is limited. Recall that an attacker can pretend to be a normal user and use the encoder E of the victim model S as a black box to obtain watermarked images with known messages. While this is realistic because many real-world service providers (e.g., StegAI[1] and Digimarc[2]) allow users to upload an image and specify the watermark message, acquiring a large number of watermarked images can be costly since such services may be pay-per-use. Hence, we evaluate the attack effectiveness of our methods under this constrained scenario by varying the number of accessible watermarked images.

Fig. 5 shows the results of Δ_{ACC} on COCO when attacking HD, UDH, ROS and CRIW, where we train the noise images and the selective neural network with 10,000, 7,000, 4,000, and 1,000 watermarked images, respectively, and train the surrogate model with the same number of watermarked images. We adopt the same experimental settings as in Fig. 4 and show the results of the baseline methods as reference. Again, we cannot report the results of WMAttacker, Diffpure, RinseDiff and RinseVAE because the maximum average PSNR they can achieve is much lower than that achieved by our methods.

We can see from Fig. 5 that as the number of watermarked images decreases, the Δ_{ACC} of our methods is reduced accordingly. However, with only 1,000 watermarked images, our methods still outperform the baseline methods in most cases. Since obtaining 1,000 watermarked images is economically feasible (e.g., costing only 100 USD on StegAI), our methods are cost-effective and maintain superior attack performance in this real-world scenario.

[1] https://steg.ai
[2] https://www.digimarc.com

VII. Conclusion

In this work, we introduce a novel fast and effective overwrite attack (FEOA) against DNN-based image watermarking models. FEOA overwrites the true watermark messages of many watermarked images by simply adding the same noise image to them. This noise image is used to attack different watermarked images without any re-training or fine-tuning, thus enabling fast and effective overwrite attacks. We also extend FEOA to its adaptive version named AFEOA to further improve attack performance. Extensive experimental results demonstrate the excellent performance of FEOA and AFEOA.

References

[1] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 657–672.

[2] C. Zhang, P. Benz, A. Karjauv, G. Sun, and I. S. Kweon, "Udh: Universal deep hiding for steganography, watermarking, and light field messaging," *Advances in Neural Information Processing Systems*, pp. 10 223–10 234, 2020.

[3] X. Zhong, P.-C. Huang, S. Mastorakis, and F. Y. Shih, "An automated and robust image watermarking scheme based on deep neural networks," *IEEE Transactions on Multimedia*, pp. 1951–1961, 2020.

[4] X. Luo, Y. Li, H. Chang, C. Liu, P. Milanfar, and F. Yang, "Dvmark: a deep multiscale framework for video watermarking," *IEEE Transactions on Image Processing*, 2023.

[5] Y. Tang, C. Wang, S. Xiang, and Y.-m. Cheung, "A robust reversible watermarking scheme using attack-simulation-based adaptive normalization and embedding," *IEEE Transactions on Information Forensics and Security*, 2024.

[6] C. Qin, X. Li, Z. Zhang, F. Li, X. Zhang, and G. Feng, "Print-camera resistant image watermarking with deep noise simulation and constrained learning," *IEEE Transactions on Multimedia*, 2023.

[7] P. Fernandez, A. Sablayrolles, T. Furon, H. Jégou, and M. Douze, "Watermarking images in self-supervised latent spaces," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 3054–3058.

[8] X. Luo, R. Zhan, H. Chang, F. Yang, and P. Milanfar, "Distortion agnostic deep watermarking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 548–13 557.

[9] Z. Jia, H. Fang, and W. Zhang, "Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression," in *Proceedings of the ACM International Conference on Multimedia*, 2021, pp. 41–49.

[10] M. Tancik, B. Mildenhall, and R. Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2117–2126.

[11] H. Fang, Z. Jia, Y. Qiu, J. Zhang, W. Zhang, and E.-C. Chang, "De-end: decoder-driven watermarking network," *IEEE Transactions on Multimedia*, pp. 7571–7581, 2022.

[12] T. Bui, S. Agarwal, N. Yu, and J. Collomosse, "Rosteals: Robust steganography using autoencoder latent space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 933–942.

[13] H. Fang, Z. Jia, H. Zhou, Z. Ma, and W. Zhang, "Encoded feature enhancement in watermarking network for distortion in real scenes," *IEEE Transactions on Multimedia*, pp. 2648–2660, 2022.

[14] P. Fernandez, G. Couairon, H. Jégou, M. Douze, and T. Furon, "The stable signature: Rooting watermarks in latent diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 466–22 477.

[15] Y. Wen, J. Kirchenbauer, J. Geiping, and T. Goldstein, "Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust," *arXiv preprint arXiv:2305.20030*, 2023.

[16] Z. Jiang, M. Guo, Y. Hu, J. Jia, and N. Z. Gong, "Certifiably robust image watermark," *arXiv preprint arXiv:2407.04086*, 2024.

[17] N. Yu, V. Skripniuk, S. Abdelnabi, and M. Fritz, "Artificial fingerprinting for generative models: Rooting deepfake attribution in training data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 448–14 457.

[18] H. Zhong, J. Chang, Z. Yang, T. Wu, P. C. Mahawaga Arachchige, C. Pathmabandu, and M. Xue, "Copyright protection and accountability of generative ai: Attack, watermarking and attribution," in *Companion Proceedings of the ACM Web Conference 2023*, 2023, pp. 94–98.

[19] S. Ranjbar Alvar, M. Akbari, D. Yue, and Y. Zhang, "Nft-based data marketplace with digital watermarking," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 4756–4767.

[20] K. Kenthapadi, H. Lakkaraju, and N. Rajani, "Generative ai meets responsible ai: Practical challenges and opportunities," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 5805–5806.

[21] M. Sag, "Copyright safety for generative ai," *Forthcoming in the Houston Law Review*, 2023.

[22] S. Shoker, A. Reddie, S. Barrington, M. Brundage, H. Chahal, M. Depp, B. Drexel, R. Gupta, M. Favaro, J. Hecla *et al.*, "Confidence-building measures for artificial intelligence: Workshop proceedings," *arXiv preprint arXiv:2308.00862*, 2023.

[23] Z. Jiang, J. Zhang, and N. Z. Gong, "Evading watermark based detection of ai-generated content," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 1168–1181.

[24] G. Li, Y. Chen, J. Zhang, J. Li, S. Guo, and T. Zhang, "Warfare:breaking the watermark protection of ai-generated content," *arXiv preprint arXiv:2310.07726*, 2023.

[25] N. Lukas, A. Diaa, L. Fenaux, and F. Kerschbaum, "Leveraging optimization for adaptive attacks on image watermarks," *The International Conference on Learning Representations*, 2024.

[26] M. Saberi, V. S. Sadasivan, K. Rezaei, A. Kumar, A. Chegini, W. Wang, and S. Feizi, "Robustness of ai-image detectors: Fundamental limits and practical attacks," *arXiv preprint arXiv:2310.00076*, 2023.

[27] X. Zhao, K. Zhang, Z. Su, S. Vasan, I. Grishchenko, C. Kruegel, G. Vigna, Y.-X. Wang, and L. Li, "Invisible image watermarks are provably removable using generative ai," *arXiv preprint arXiv:2306.01953*, 2023.

[28] A. Kassis and U. Hengartner, "Unmarker: A universal attack on defensive watermarking," *arXiv preprint arXiv:2405.08363*, 2024.

[29] Y. Hu, Z. Jiang, M. Guo, and N. Gong, "A transfer attack to image watermarks," *arXiv preprint arXiv:2403.15365*, 2024.

[30] C. Wang, Q. Hao, S. Xu, B. Ma, Z. Xia, Q. Li, J. Li, and Y.-Q. Shi, "Rdiwan: Residual dense based imperceptible watermark attack network," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 7460–7472, 2022.

[31] B. An, M. Ding, T. Rabbani, A. Agrawal, Y. Xu, C. Deng, S. Zhu, A. Mohamed, Y. Wen, T. Goldstein *et al.*, "Benchmarking the robustness of image watermarks," *arXiv preprint arXiv:2401.08573*, 2024.

[32] R. Ma, M. Guo, Y. Hou, F. Yang, Y. Li, H. Jia, and X. Xie, "Towards blind watermarking: Combining invertible and non-invertible mechanisms," in *Proceedings of the ACM International Conference on Multimedia*, 2022, pp. 1532–1542.

[33] X. Xian, G. Wang, X. Bi, J. Srinivasa, A. Kundu, M. Hong, and J. Ding, "Raw: A robust and agile plug-and-play watermark framework for ai-generated images with provable guarantees," *arXiv preprint arXiv:2403.18774*, 2024.

[34] Y. Cui, J. Ren, H. Xu, P. He, H. Liu, L. Sun, Y. Xing, and J. Tang, "Diffusionshield: A watermark for copyright protection against generative diffusion models," *arXiv preprint arXiv:2306.04642*, 2023.

[35] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.

[36] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar, "Diffusion models for adversarial purification," *arXiv preprint arXiv:2205.07460*, 2022.

[37] Y. Hu, Z. Jiang, M. Guo, and N. Gong, "Stable signature is unstable: Removing image watermark from diffusion models," *arXiv preprint arXiv:2405.07145*, 2024.

[38] P. Yang, H. Ci, Y. Song, and M. Z. Shou, "Steganalysis on digital watermarking: Is your defense truly impervious?" *arXiv preprint arXiv:2406.09026*, 2024.

[39] A. A. Goldstein, "Convex programming in hilbert space," 1964.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[41] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[42] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy*, 2017, pp. 39–57.

[43] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1765–1773.

[44] C. Zhang, P. Benz, A. Karjauv, and I. S. Kweon, "Data-free universal adversarial perturbation and black-box attack," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7868–7877.

[45] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[46] M. Tsivgoulis, T. Papastergiou, and V. Megalooikonomou, "An improved squeezenet model for the diagnosis of lung cancer in ct scans," *Machine Learning with Applications*, p. 100399, 2022.

[47] A. S. Agoes, Z. Hu, and N. Matsunaga, "Fine tuning based squeezenet for vehicle classification," in *Proceedings of the International Conference on Advances in Image Processing*, 2017, pp. 14–18.

[48] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[49] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv preprint arXiv:1611.00712*, 2016.

[50] I. A. Huijben, W. Kool, M. B. Paulus, and R. J. Van Sloun, "A review of the gumbel-max trick and its extensions for discrete stochasticity in machine learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1353–1371, 2022.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[53] K. Navas, M. C. Ajay, M. Lekshmi, T. S. Archana, and M. Sasikumar, "Dwt-dct-svd based watermarking," in *International Conference on Communication Systems Software and Middleware and Workshops*, 2008, pp. 271–274.

[54] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 740–755.

[55] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[56] P. Sharma, N. Ding, S. Goodman, and R. Soricut, "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 2556–2565.

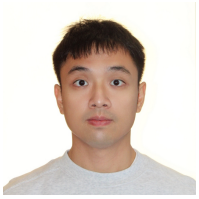
[57] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in Neural Information Processing Systems*, 2017.

[58] P. Pernias, D. Rampas, M. L. Richter, C. J. Pal, and M. Aubreville, "Würstchen: An efficient architecture for large-scale text-to-image diffusion models," *arXiv preprint arXiv:2306.00637*, 2023.

[59] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[61] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.



Shaoxin Li received the B.Eng. degree in information engineering from Xihua University, China, in 2016. He received the M.Eng. degree in signal and information processing from Southwest University, China, in 2019. He is currently a Ph.D. student in the College of Computer Science, Chongqing University, China. He was a visiting student in the Department of Computing and Software, McMaster University, Canada, during 2021 - 2023. His research interests include data mining, trustworthy artificial intelligence and applied machine learning.

Xiaofeng Liao (Fellow, IEEE) received the B.Sc. and M.Sc. degrees in mathematics from Sichuan University, Chengdu, China, in 1986 and 1992, respectively, and the Ph.D. degree in circuits and systems from the University of Electronic Science and Technology of China, Chengdu, in 1997. From 1999 to 2012, he was a Professor with Chongqing University, Chongqing, China. From July 2012 to July 2018, he was a Professor and the Dean of the College of Electronic and Information Engineering, Southwest University, Chongqing. He is currently a Professor and the Dean of the College of Computer Science, Chongqing University. He is also a Yangtze River Scholar of the Ministry of Education of China, Beijing, China. From November 1997 to April 1998, he was a Research Associate with The Chinese University of Hong Kong, Hong Kong. From October 1999 to October 2000, he was a Research Associate with The City University of Hong Kong, Hong Kong, where he was a Senior Research Associate from March 2001 to June 2001 and from March 2002 to June 2002, where he was a Research Fellow from March 2006 to April 2007. He holds five patents, and published four books and over 300 international journal and conference papers. His current research interests include optimization and control, machine learning, neural networks, bifurcation and chaos, and cryptography.

Qiqi Zhang received the B.Sc. degree in math and the M.Sc. degree in computer science from McMaster University, Canada, in 2021 and 2023, respectively. She is currently a Ph.D. student in the Department of Computing and Software, McMaster University. Her research interests include data mining, trustworthy artificial intelligence, and machine learning algorithms' robustness and explainability.

Yuanqi Xue received her B.A.Sc. degree in computer science from McMaster University, Canada, in 2024. She is currently pursuing an M.Sc. in the Department of Computing and Software, McMaster University. Her research interests include trustworthy artificial intelligence and unsupervised learning.

Lingyang Chu is an assistant professor at the Department of Computing and Software, McMaster University, Canada. He was a postdoctoral fellow in the School of Computing Science, Simon Fraser University, Canada, during 2015 - 2018. He received the doctoral degree from the University of Chinese Academy of Sciences, China, in 2015. His research interests include data mining, trustworthy artificial intelligence and applied machine learning. He is an associate editor of ACM Transactions on Knowledge Discovery from Data.

APPENDIX A
CONTROLLING THE AMOUNT OF NOISE

In this section, we introduce how to control the amount of noise for each attack method in our experiments as follows.

FEOA-WB, FEOA-BB, AFEOA-WB and AFEOA-BB overwrite the true message carried by a watermarked image by adding a noise image to it. The amount of noise is controlled by the threshold ξ, which limits the L_2-norm of the noise image. We use $\xi \in \{100, 200, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000\}$ for SSL, ROS, CRIW, SS and TR, and we use $\xi \in \{50, 100, 200, 400, 600, 800, 1000, 1200, 1400, 1600\}$ for HD, UDH and DDS.

WMAttacker [27] attacks a watermarked image by first adding a Gaussian noise sampled from $\mathcal{N}(0, \sigma^2)$ to each entry in the embedding of the watermarked image and then reconstructing the image from the noised embedding by Stable Diffusion [35]. As demonstrated in [27], WMAttacker controls the amount of Gaussian noise by the standard deviation σ. We use $\sigma \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$.

Diffpure [26] attacks a watermarked image by using a pre-trained diffusion model [36] to denoise the watermarked image. As shown in [26], Diffpure indirectly controls the amount of noise by the diffusion purification step t. We use $t \in \{0.05, 0.10, 0.15, 0.20, 0.25, 0.30\}$.

RinseDiff [31] attacks a watermarked image by repeating Z times the process of noising the embedding of the image and reconstructing the image by Stable Diffusion [35]. According to [31], Z mainly controls the amount of the added noise. We use $Z \in \{1, 2, 3\}$.

RinseVAE [31] attacks a watermarked image by repeating Z times the process of using a pre-trained variational autoencoder (VAE) [59] to encode and decode the watermarked image. According to [31], Z mainly controls the amount of the added noise. We use $Z \in \{1, 2, 3\}$.

SBR [38] attacks a watermarked image by subtracting the watermark pattern δ_w extracted by SBR from a set of watermarked images. According to [38], SBR controls the amount of noise by multiplying δ_w by a factor of M. We use $M \in \{0.5, 1, 1.5, 2.0, 2.5, 3.0\}$.

Gaussian noise [8], [17], [27] adds a random Gaussian noise sampled from $\mathcal{N}(0, \sigma^2)$ to each pixel of a watermarked image. The standard deviation σ controls the amount of noise. We use $\sigma \in \{0.02, 0.04, 0.06, 0.08, 0.10\}$.

Gaussian blur [1], [2], [7], [27] blurs a watermarked image by convolving it with a Gaussian kernel to smoothen the signal of the embedded message. The Gaussian kernel has a kernel size s and a standard deviation σ . We set $s = 5$ and use $\sigma \in \{0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$ to control the amount of added noise.

JPEG [9], [27], [32] is a common lossy image compression technique. It has a parameter called quality factor Q, which affects the amount of noise. We use $Q \in \{10, 20, 30, 40, 50\}$.

APPENDIX B
ADDITIONAL ATTACK PERFORMANCE

Here, we introduce the additional results on the attack performance of different attack methods, such as the results of Δ_{ACC} and average PSNR on ImageNet, and the results of Δ_{ACC} and FID on both COCO and ImageNet.

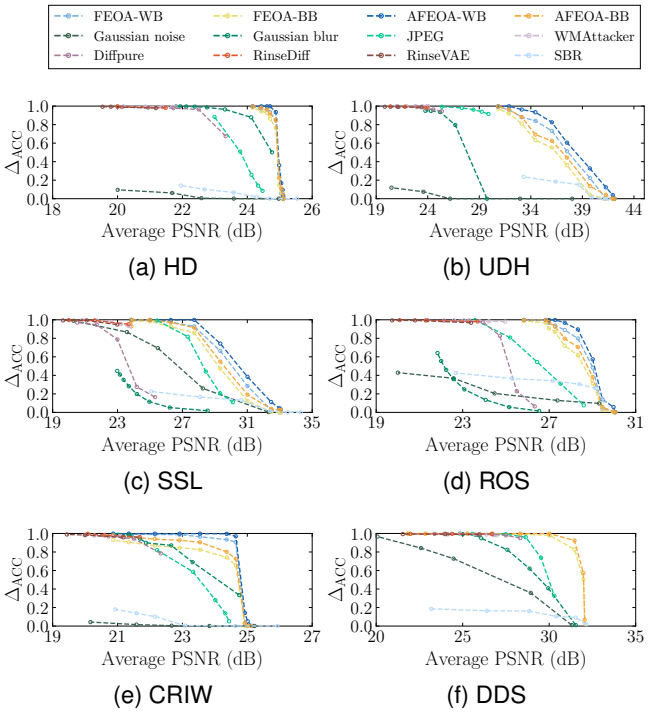


Fig. 6. The Δ_{ACC} and average PSNR on ImageNet. The caption of each subfigure shows the name of victim model.

Fig. 6 shows the results of Δ_{ACC} and average PSNR on ImageNet, which complement the results previously shown in Fig. 2. We cannot report the results of attacking SS and TR on ImageNet, because ImageNet does not provide image captions for SS and TR to generate watermarked images. We can observe from Fig. 6 that our methods achieve the highest Δ_{ACC} while maintaining the largest average PSNR, which demonstrates the superior attack performance of our methods.

Fig. 7 and Fig. 8 present the results of Δ_{ACC} and FID on COCO and ImageNet, respectively. In each figure, we report the Δ_{ACC} and FID of each attack method when using different amounts of noise. For each specific amount of noise, we produce one pair of Δ_{ACC} and FID, thus drawing one point in the figure. A point that is closer to the upper left corner of the figure indicates higher attack effectiveness (i.e., a higher Δ_{ACC}) and better quality of attacked images (i.e., a lower FID), thereby implying better attack performance. As shown in Fig. 7 and Fig. 8, our methods outperform the baseline methods by achieving the highest Δ_{ACC} while maintaining the lowest FID. This further demonstrates the good attack performance of our methods.

APPENDIX C
TOTAL ATTACK TIME ON IMAGENET

In this section, we present the additional results on the attack efficiency. Table VIII reports the total attack time of different attack methods on ImageNet, where the amount of noise for each attack method is set to achieve the largest Δ_{ACC} in Fig. 6. We cannot report the results of attacking SS and TR on ImageNet since it does not provide image captions for SS and

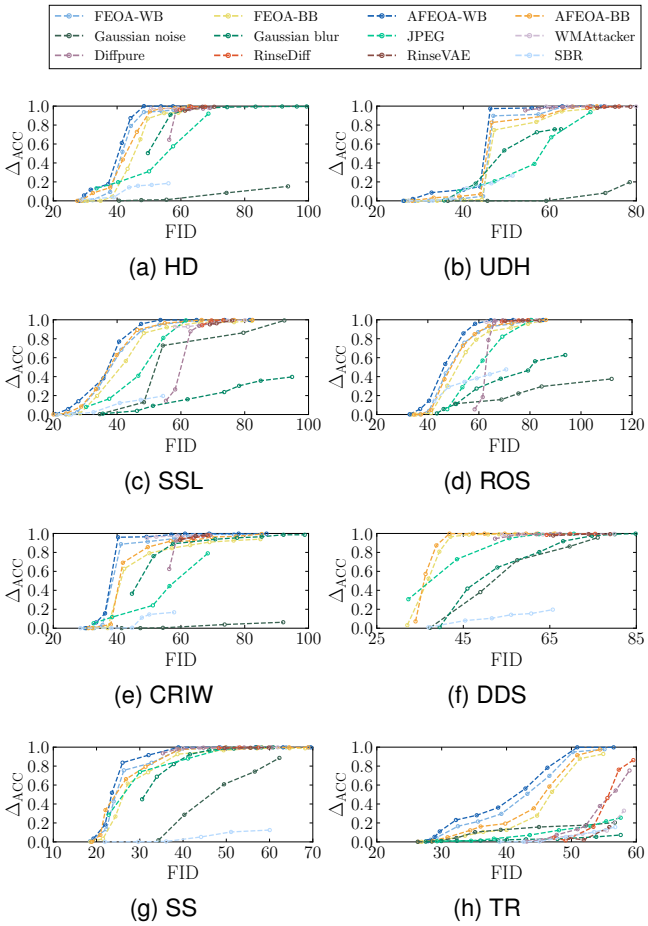


Fig. 7. The Δ_{ACC} and FID on COCO. The caption of each subfigure shows the name of victim model.

TABLE VIII
THE TOTAL ATTACK TIME (MILLISECONDS) ON IMAGENET.

# attacked images	1,000						5,000	
Victim models	HD	UDH	SSL	ROS	CRIW	DDS	SS	TR
FEOA-WB	**14**	**14**	**14**	**14**	**13**	n/a	n/a	n/a
FEOA-BB	**13**	**13**	**14**	**18**	**14**	**14**	n/a	n/a
AFEOA-WB	2,424	2,360	2,334	2,868	2,482	n/a	n/a	n/a
AFEOA-BB	2,569	2,364	2,348	2,832	2,496	2,502	n/a	n/a
Gaussian noise	112	114	116	135	115	119	n/a	n/a
Gaussian blur	207	211	219	248	209	216	n/a	n/a
JPEG	12,538	12,577	12,542	13,553	12,584	12,520	n/a	n/a
WMAttacker	51,484	51,220	51,929	58,539	51,530	51,349	n/a	n/a
Diffpure	85,813	85,258	84,920	97,656	85,894	85,530	n/a	n/a
RinseDiff	146,860	148,323	143,698	164,923	152,534	144,806	n/a	n/a
RinseVAE	75,295	75,816	76,118	85,230	74,636	74,915	n/a	n/a
SBR	**14**	**14**	**15**	**18**	**16**	**14**	n/a	n/a

TR to generate watermarked images. Table VIII complements the results previously reported in Table I, showing that the results on ImageNet are consistent with those on COCO, which further demonstrates the high attack efficiency of our methods.

APPENDIX D
THE EFFECT OF k ON IMAGENET

Here, we introduce the additional results on the analysis of the number of noise images k in Fig. 9, which complement the results previously shown in Fig. 3. We follow the experimental settings in Section VI-E. Again, we cannot report the results

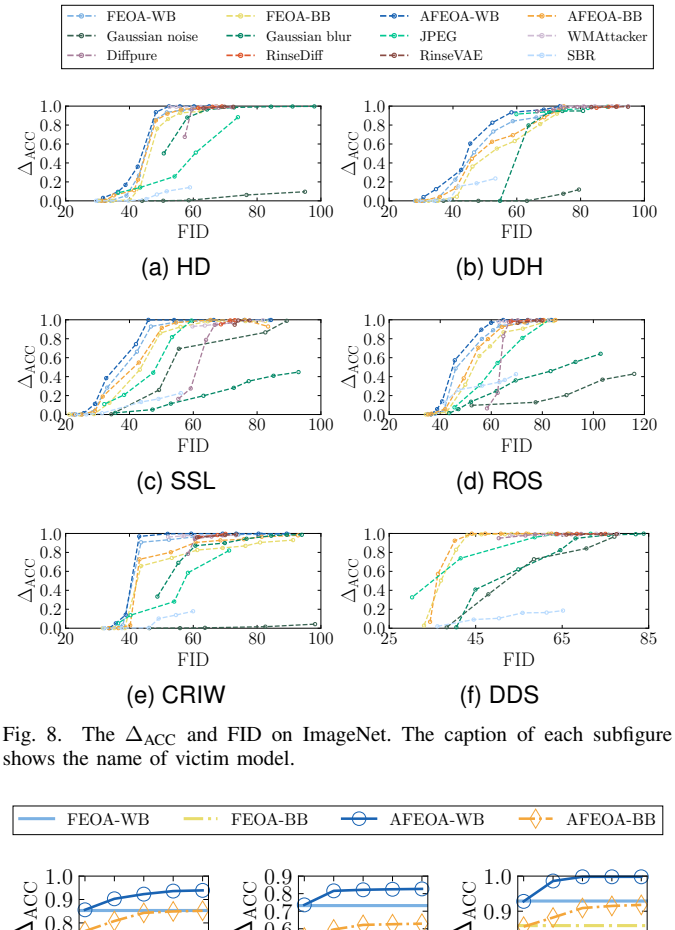


Fig. 8. The Δ_{ACC} and FID on ImageNet. The caption of each subfigure shows the name of victim model.

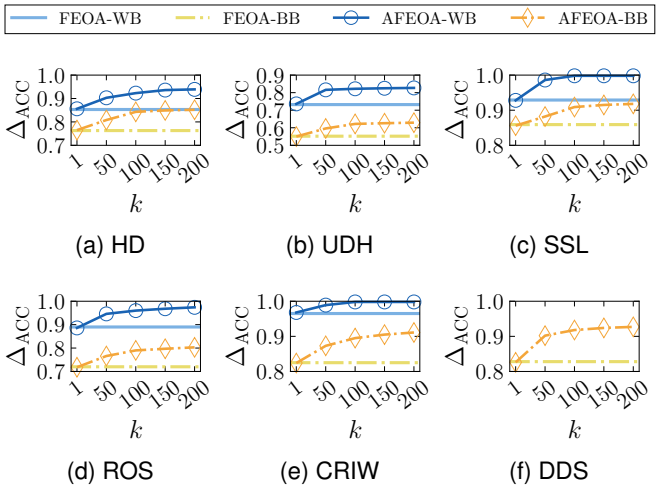


Fig. 9. The Δ_{ACC} for $k \in \{1, 50, 100, 150, 200\}$ on ImageNet.

of attacking SS and TR on ImageNet, as it does not provide image captions for SS and TR to generate watermarked images. The additional results in Fig. 9 are similar to the previous results shown in Fig. 3, thus enhancing our conclusion about AFEOA in Section VI-E. This further demonstrates that the training process of AFEOA is stable and consistent on different image watermarking models and datasets.

APPENDIX E
THE LOWER BOUND OF PSNR

In this section, we empirically validate the lower bound of PSNR given by (1). Table IX reports the results of our methods on COCO and ImageNet when using $\xi = 800$ for HD, UDH and DDS and using $\xi = 2000$ for SSL, ROS, CRIW, SS

TABLE IX
THE LOWER BOUND OF PSNR AND THE EMPIRICAL PSNR ON COCO AND IMAGENET.

Dataset	Attack	PSNR	HD	UDH	SSL	ROS	CRIW	DDS	SS	TR
COCO	FEOA-WB	LB	36.9	36.9	29.0	35.0	29.0	n/a	35.0	35.0
		Empirical	36.9	37.0	29.0	35.0	29.0	n/a	35.0	35.1
	FEOA-BB	LB	36.9	36.9	29.0	35.0	29.0	36.9	35.0	35.0
		Empirical	36.9	36.9	29.0	35.0	29.0	36.9	35.0	35.0
	AFEOA-WB	LB	36.9	36.9	29.0	35.0	29.0	n/a	35.0	35.0
		Empirical	37.0	37.0	29.0	35.0	29.0	n/a	35.0	35.0
	AFEOA-BB	LB	36.9	36.9	29.0	35.0	29.0	36.9	35.0	35.0
		Empirical	36.9	36.9	29.0	35.0	29.0	36.9	35.0	35.0
ImageNet	FEOA-WB	LB	36.9	36.9	29.0	35.0	29.0	n/a	n/a	n/a
		Empirical	36.9	36.9	29.0	35.0	29.0	n/a	n/a	n/a
	FEOA-BB	LB	36.9	36.9	29.0	35.0	29.0	36.9	n/a	n/a
		Empirical	37.0	36.9	29.0	35.1	29.0	37.0	n/a	n/a
	AFEOA-WB	LB	36.9	36.9	29.0	35.0	29.0	n/a	n/a	n/a
		Empirical	36.9	36.9	29.0	35.0	29.0	n/a	n/a	n/a
	AFEOA-BB	LB	36.9	36.9	29.0	35.0	29.0	36.9	n/a	n/a
		Empirical	36.9	36.9	29.1	35.0	29.0	36.9	n/a	n/a

TABLE X
THE PERCENTAGE OF WATERMARKED IMAGES FOR WHICH h_β SUCCEEDS IN SELECTING THE MOST EFFECTIVE NOISE IMAGE.

Dataset	Attack	HD	UDH	SSL	ROS	CRIW	DDS	SS	TR
COCO	AFEOA-WB	0.91	0.94	0.92	0.94	0.95	n/a	0.93	0.90
	AFEOA-BB	0.84	0.88	0.85	0.82	0.90	0.89	0.88	0.85
ImageNet	AFEOA-WB	0.93	0.90	0.93	0.92	0.96	n/a	n/a	n/a
	AFEOA-BB	0.86	0.83	0.84	0.81	0.88	0.90	n/a	n/a

and TR. The other experimental settings are the same as in Section VI-E. Here, "LB" means the theoretical PSNR lower bound computed by the right side of (1), and "Empirical" means the empirical minimum value of PSNR$(I_w, I_w + \delta)$ for all the watermarked images I_w. As shown in Table IX, the empirical minimum value of PSNR$(I_w, I_w + \delta)$ is always no smaller than the theoretical PSNR lower bound. This demonstrates the correctness of (1).

APPENDIX F
VALIDITY OF THE SELECTIVE NEURAL NETWORK

In this section, we empirically verify the validity of the selective neural network h_β in selecting the most effective noise image in P for each watermarked image. Specifically, for each watermarked image I_w in the testing dataset, we first use h_β to select the noise image in P corresponding to the highest probability output by $h_\beta(I_w)$, denoted as δ_{i^*}. Then, we compute the function value of $L(F(I_w+\delta_{i^*}), w)$ and compare it with the function values when using the other noise images in P to attack I_w. If δ_{i^*} produces the largest function value among all the noise images in P, then we say h_β successfully selects the most effective noise image in P for I_w.

Table X reports the percentage of watermarked images for which h_β successfully selects the most effective noise image for each of them, where we follow the experimental settings in Section VI-E. We do not report the results of FEOA-WB and FEOA-BB since they only train a single noise image and do not use h_β. As shown in Table X, when attacking different victim models in the white-box setting by applying AFEOA-WB, h_β successfully selects the most effective noise image for over 90% watermarked images. This well demonstrates the validity of h_β. In comparison, the percentage is slightly lower when applying AFEOA-BB in the black-box setting. This is due to the performance discrepancy between the victim model and the surrogate model F_S, which means that the most effective noise image selected by h_β to attack F_S is not necessarily also the most effective nose image in attacking the victim model. Nevertheless, h_β still successfully selects the most effective noise image for over 81% watermarked images.

Another interesting finding is that, in some cases, the percentage of watermarked images for which h_β succeeds in selecting the most effective noise image is lower than the corresponding Δ_{ACC}. For instance, when applying AFEOA-WB to attack SS on COCO, the percentage of watermarked images reported in Table X is 0.93, whereas the corresponding Δ_{ACC} is 0.99. This is because even though h_β fails to select the most effective noise image for a watermarked image I_w, the selected noise image may still be able to successfully attack I_w due the high effectiveness of each noise image in P.

APPENDIX G
CASE STUDY

In this section, we conduct a case study to show the attacked images produced by each attack method, as well as the noise images used by our methods to perform the attacks.

Fig. 10(a) and Fig. 10(b) show the results of attacked images on COCO and ImageNet, respectively. Fig. 10(b) does not include the results on SS and TR, because ImageNet does not provide image captions for SS and TR to produce watermarked images. Each column in Fig. 10 shows the results on one original image for a victim model, where rows 1 and 2 display the original image and the watermarked image, respectively, and rows 3 to 14 show the attacked images produced by different attack methods. The attacked images generated by our methods are framed by blue and red rectangles in Fig. 10(a) and Fig. 10(b), respectively. We also show the noise images used to produce the attacked images within the blue and red rectangles in Fig. 11(a) and Fig. 11(b), respectively. Each noise image is visualized in the same way as in [43].

For each attack method, we set the noise amount to achieve the highest PSNR while obtaining a Δ_{ACC} of at least 0.8. The numbers below the attacked images show the PSNR achieved by each attack method. An underlined PSNR means the corresponding attack method cannot achieve a $\Delta_{\text{ACC}} \geq 0.8$. In this case, we report the highest PSNR when the attack method can successfully attack the watermarked image. The bold green-colored PSNR in each column of Fig. 10 indicates the highest PSNR (i.e., the best image quality) of the attacked images when attacking the corresponding victim model in the same column.

As shown in Fig. 10, most of the attacked images produced by Gaussian noise, Gaussian blur and JPEG have a small PSNR, and the quality of these images is not very satisfactory. In addition, the attacked images produced by WMAttacker, Diffpure, RinseDiff and RinseVAE tend to be over-smoothed and thus also have a small PSNR. Compared with the baseline methods, our methods achieve the best image quality and a higher PSNR. This is because the L_2-norm of a noise image used in our methods is upper-bounded by the threshold ξ,

which effectively limits the damage to image quality caused by adding the noise image.

Another interesting finding is that, as shown in columns 3 and 7 of Fig. 11(a), the noise images for attacking HD and CRIW exhibit similar patterns (e.g., diagonal lines). A similar phenomenon also appears in columns 1 and 5 of Fig. 11(b). We explain the reasons for this phenomenon as follows. First, according to [16], CRIW adopts HD as the base watermarking model. This is, CRIW uses the original encoder of HD as its encoder with only a few modifications and it "smooths" the original decoder of HD as its decoder by applying the randomized smoothing technique [62]. Second, since HD and CRIW use virtually the same encoder, the watermarked images they generate have a very similar distribution of watermark patterns. Third, although the decoders of HD and CRIW do not extract watermarks in exactly the same way, they both aim at recognizing the watermark patterns embedded in the watermarked images. Since the watermark patterns embedded by HD and CRIW have a similar distribution, their decoders work in a similar way to recognize these watermark patterns. Last, when applying our methods, the noise images used to attack HD and CRIW are not only trained on the watermarked images with a similar distribution of watermark patterns, but are also trained to disrupt the decoders that recognize the watermark patterns in a similar way. Thus, the mechanisms of effect of these noise images to attack HD and CRIW are similar, which results in their similar visual patterns.

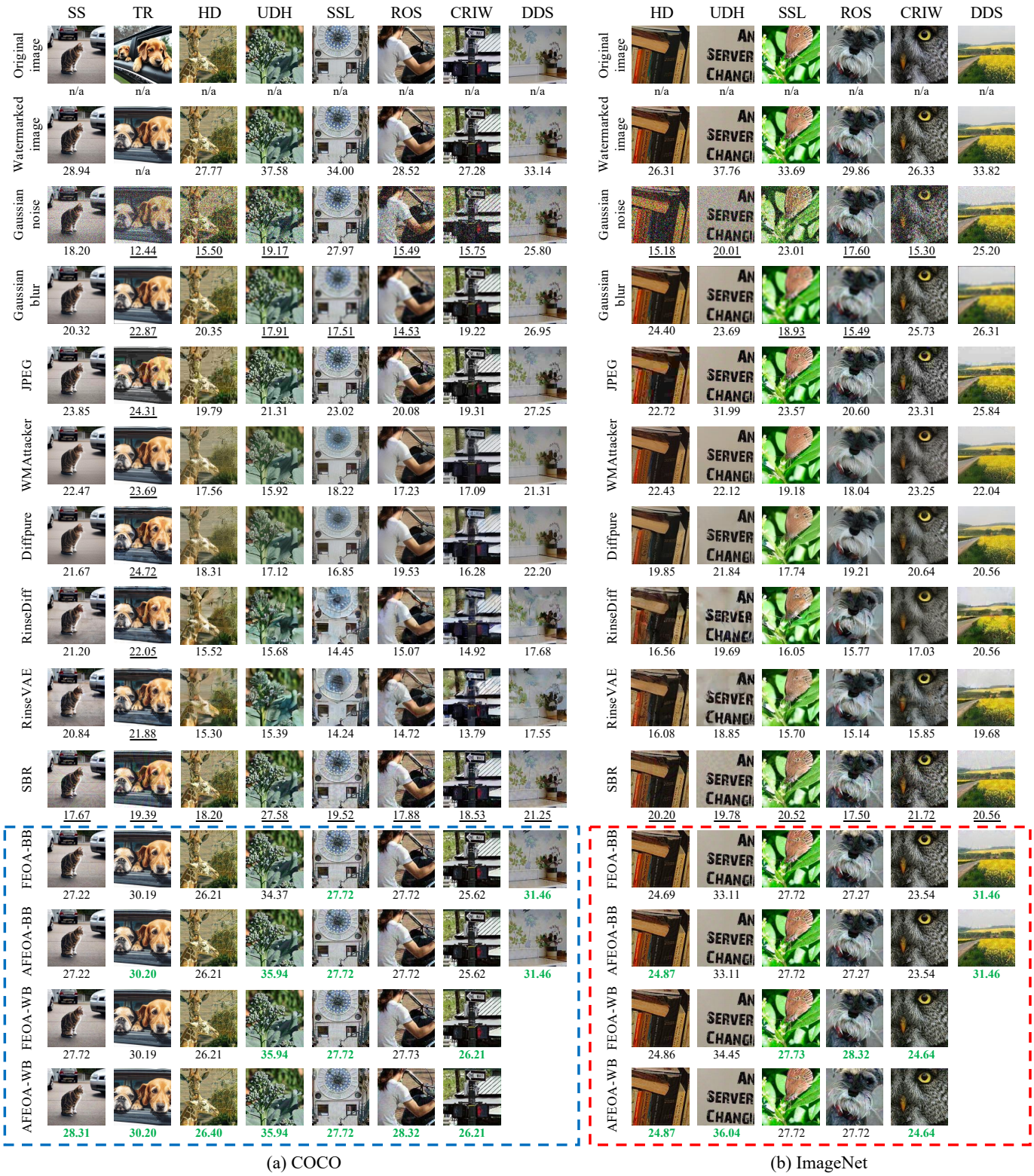


Fig. 10. The case study results on COCO and ImageNet. All the watermarked images are successfully attacked by corresponding attacks. FEOA-WB and AFEOA-WB are not applicable to attack DDS, because the decoder of DDS is not differentiable.

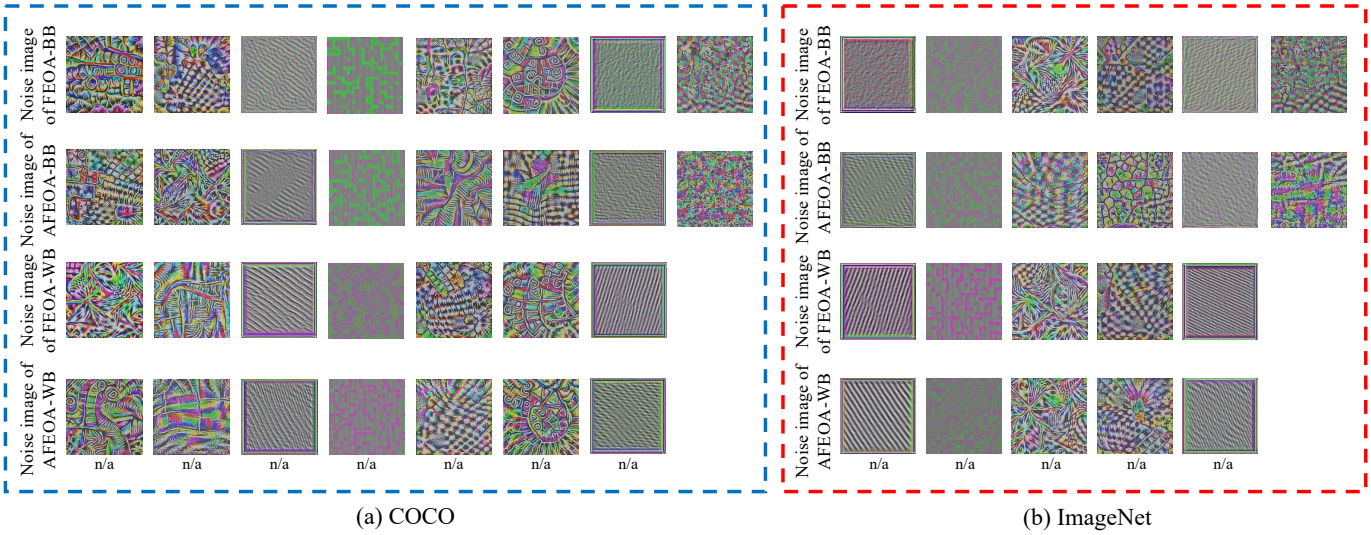


Fig. 11. The noise images corresponding to the attacked images generated by our methods in Fig. 10.