Model Extraction Attack and Its Countermeasure for Denoising Diffusion Implicit Models

Hayato Shoji* and Kazuaki Nakamura*
* Tokyo University of Science, Japan
E-mail: 4625518@ed.tus.ac.jp. nakamura.kazuaki@rs.tus.ac.jp

***Abstract*—Recently, the threat of cyber attacks against machine learning models has been increasing. Typical examples include Model Extraction Attack (MEA), which steals the functionality of a victim model by creating its clone model that has almost the same functionality. Thus, the literature has studied MEA and its defense methods, mainly focusing on image recognition models. However, no existing studies evaluate the risk of MEA on diffusion-based image generation models, despite the recent advances and widespread use of image generation AI services powered by diffusion models. In this paper, we first demonstrate the feasibility of MEA on DDIM, one of the most common diffusion-based image generation models. Then, as a countermeasure, we propose a defense method that detects clone models of DDIM. In the proposed method, we add a small number of out-of-distribution images, referred to as "marking images", to the training dataset of a victim DDIM model. This technique provides the property of occasionally generating marking images for the victim model. This property works as a watermark and is inherited by the clone models, being used as a clue for detecting them. In the results of our experiments conducted on face, fruit, and church image datasets, the proposed defense method can correctly detect all clone models without seriously degrading the usability of victim DDIM models.**

I. Introduction

In recent years, cloud services powered by machine learning models have been actively developed in various fields, such as image recognition, image generation, and language translation. On the other hand, the threat of cyber attacks against these machine learning models is also increasing. Typical examples include Model Extraction Attack (MEA), which steals a victim model's functionality by maliciously training its clone model. Specifically, adversaries send a series of queries to the victim model and collect the corresponding outputs, which are then used as a training dataset to train a clone model that has almost the same functionality as the original victim model. The conventional studies of MEA have focused on an image recognition model as their target victim model [1], [2], [3], [4], [5]. However, with the recent development and spread of generative AI services, adversaries whose interest shifts from image recognition models to image generation models will increase. Hence, it is urgent to develop a defense method against MEA on image generation models.

So far, only a few existing studies have focused on MEA that targets image generation models. Hu et al. [6] proposed an MEA method on Generative Adversarial Networks (GAN) [7]. Besides, Liu et al. [8] proposed an MEA method that targets Variational Auto-Encoders (VAE) [9] and PixcelCNN [10] in addition to GAN. These studies demonstrated the feasibility of MEA on GAN and VAE. In addition, they also provided a defense approach against their proposed MEA methods, where a victim model's owner sets an upper limit on the number of query accesses to prevent adversaries from collecting a sufficient amount of data for training a clone model.

However, existing MEA studies described above have two issues. **First, they do not focus on diffusion models; no existing studies have targeted MEA on diffusion models** to the best of our knowledge. Diffusion models attract more and more attention as an image generation model that is capable of generating higher quality images than state-of-the-art GAN [11]. Indeed, modern text-driven image generation services, including Stable Diffusion, DALL-E, and Imagen, are all based on diffusion models. Therefore, analyzing the risk of MEA on diffusion models is urgent. **Second, existing studies do not consider methods for detecting clone models despite their importance**. There are two types of defense strategies against MEA: prevention and detection. The former is to prevent adversaries from training a clone model by altering the generation results or prohibiting query access, as Hu [6] and Liu [8] did, while the latter is to detect and disable clone models ex post facto. We believe that combining these two strategies is important for a robust defense against MEA. Hence, clone detection methods need to be explored.

In this paper, we focus on unconditional Denoising Diffusion Implicit Models (DDIM) [12], which are one of the most common diffusion-based image generation models, and evaluate the feasibility of MEA against unconditional DDIM at first. Then, as a countermeasure for the MEA, we propose a method for detecting clone models of unconditional DDIM. The basic idea in our clone detection method is watermarking. In other words, we embed a watermark in a (victim) DDIM model, which is inherited by clone models and used as a clue for detecting them. To achieve this idea, we use a small set of "marking images", which are out-of-distribution images for the victim model, and add them to the victim model's training dataset (called "main dataset"). For instance, if the victim model is a face image generator, we use non-face images such as flower images or fruit images as marking images. This procedure makes the victim model occasionally output marking images. We use this property as the above watermark; the same property is expected to be inherited by clone models and helpful in detecting them. Hereafter, we refer to a set of marking images as a "marking dataset". Fig. 1 depicts the overview of the proposed method.

979-8-3315-7206-8/25/$31.00

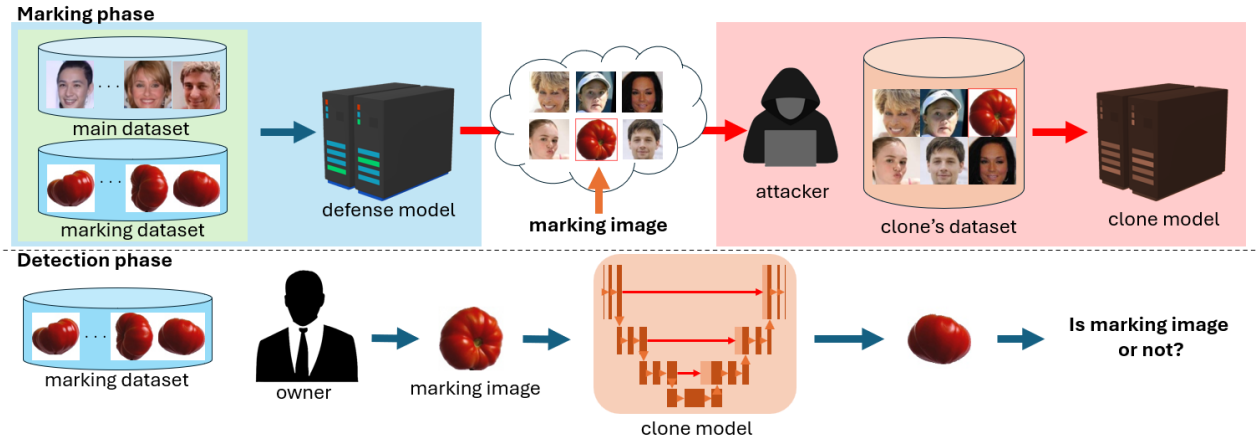

Fig. 1: Overview of the proposed method for clone detection. First, in the marking phase, we train a DDIM model using a small marking dataset in addition to the main dataset, letting the trained "defense model" acquire the property of generating "marking images" with low probability. Clone models inherit the same property since their training dataset contains a few marking images generated by the defense model. In the detection phase, we judge whether a given suspicious model is a clone or not. To this end, we input the marking images into the suspicious model and check whether they are successfully reconstructed or not.

The contributions of this paper are summarized as follows.

- This is the first work to focus on MEA against diffusion-based image generation models.
- We experimentally showed an unignorable risk of MEA against unconditional DDIM.
- We proposed a method of detecting clone models as a countermeasure against MEA and experimentally demonstrated its effectiveness.

II. Related Works

A. Model Extraction Attacks

MEA was first explored by Tramer et al. [1] in 2016. The target of MEA in their work is a recognition model, whose clone model is trained as follows. First, adversaries send input samples to a victim model as queries and get the corresponding output labels. After that, they maliciously train a clone model on the pairs of an input sample and an output label. Following this work, advanced methods for MEA focusing on image recognition models have been actively studied. Wang et al. [3] proposed an attack method that estimates hyper-parameters of a victim model under the condition that its algorithm class is known. Shi et al. [2] and Oh et al. [4] proposed MEA methods on CNN models under a black-box setting.

The first work of MEA to focus on image generation models was conducted by Hu et al. [6] in 2021. Given a victim generation model G, their goal is to obtain a clone model $\hat{G}$ whose data distribution is as similar as possible to G's data distribution. They achieved this goal for GAN-based G, demonstrating the feasibility of MEA on GAN. Besides, Liu et al. [8] showed the feasibility of MEA on PixelsCNN and VAE. However, to the best of our knowledge, no existing studies have explored MEA on diffusion models including DDIM, which we address in this paper.

B. Defence Methods against MEA

The study of defense methods against MEA also started with image recognition models. As mentioned in Section I, there are two types of defense methods: prevention and detection. For the former, Orekondy et al. [5] proposed a method to prevent the training process of clone models. Their method adds a perturbation to a victim recognition model's output, making the gradient of the loss function differ between the victim model and its clone models. For the latter, methods to embed a watermark into DNN-based image recognition models [13], [14] have been proposed. The embedded watermark is inherited by clone models and used to detect them.

In contrast, for defense against MEA on image generation models, only prevention-based approaches have been explored at present. In particular, methods to introduce an upper limit on the number of queries were proposed [6], [8]. This strategy is effective because it prevents adversaries from collecting a sufficient number of images to train a clone model. However, this strategy also has a drawback; it is vulnerable to collusion attacks, where multiple adversaries separately collect a small set of images by sending relatively few queries to a victim model and merge them into a large set to train a single clone model. To achieve a robust defense, it is important to combine prevention- and detection-based approaches. Hence, this paper proposes a detection-based defense method.

C. Unconditional Diffusion Models

An unconditional diffusion model is a generation model that does not require any prompt input and consists of two processes: forward and reverse. The forward process adds a small Gaussian noise to a given image x_0 to generate x_1 and iterates this process for x_t $(t = 1, ..., T-1)$, resulting in nearly pure Gaussian noise x_T. The reverse process starts with x_T drawn from the Gaussian distribution and iteratively estimates

x_{t-1} from x_t ($t = T, \ldots, 1$). This process gradually denoises x_T and finally generates a plausible image x_0. DDPM [15] is one of the most common diffusion models, whose reverse process obtains x_{t-1} by drawing it from

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_\theta(x_t, t), \sigma_t^2 \boldsymbol{I}) \ . \tag{1}$$

The mean of the distribution $\boldsymbol{\mu}_\theta(x_t, t)$ is calculated from x_t by a U-Net with parameter θ. More specifically, the U-Net estimates the noise component of x_t, denoted by $\epsilon_\theta^{(t)}(x_t, t)$, and subtracts it from x_t to obtain $\boldsymbol{\mu}_\theta(x_t, t)$. In the training phase, DDPM creates x_t from each training image by adding a noise ϵ_t with the forward process and minimizes the loss

$$\mathcal{L} = ||\epsilon_t - \epsilon_\theta^{(t)}(x_t, t)||^2 \tag{2}$$

to get the optimal parameter θ. DDPM is computationally expensive since it requires T steps for the reverse process. DDIM improves this drawback by reducing the number of steps; it infers $x_{\tau_{i-1}}$ from x_{τ_i} over a subsequence $(x_{\tau_0}, ..., x_{\tau_m})$, where $0 = \tau_0 < \tau_1 < \cdots < \tau_{m-1} < \tau_m = T$ and $m \ll T$.

III. Proposed Method

A. Attack Procedure

We first describe a way of conducting MEA on DDIM under the assumption that a victim DDIM model G is a black box. For such a G, an adversary uses it to collect an image set $\mathcal{Z} = \{z_1, \ldots, z_M\}$, where z_i is the i-th generated image. Then, the adversary trains his own image generation model $\hat{G}$ using $\mathcal{Z}$ as a training dataset. In this phase, any diffusion-based method can be used to train $\hat{G}$, but we assume that DDIM is employed due to its high performance and computational efficiency. The trained $\hat{G}$ is a clone model that has almost the same generation capability as G if M is sufficiently large.

B. Overview of Defense Procedure

As mentioned in Section I, our proposed defense method embeds a kind of watermark into a victim model to detect its clone models. For convenience of distinguishing from an undefended victim model, we refer to a model equipped with a watermark as a "defense model" G_{def}.

To construct the watermark, our method trains G_{def} using a "main dataset" $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ and a "marking dataset" $\widetilde{\mathcal{X}} = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_K\}$ in the marking phase, where each x_i is an in-distribution (ID) sample and each $\tilde{x}_j$ is an out-of-distribution (OOD) sample called "marking image". For example, when we train a face image generator as G_{def}, we use a face image and a fruit image as x_i and $\tilde{x}_j$, respectively. This training strategy brings a unique property to G_{def}; it occasionally generates the marking images (e.g., fruit images). These marking images are also included in a clone model's training set $\mathcal{Z}$. Note that it is difficult for adversaries to perfectly notice and remove the marking images since the size of $\mathcal{Z}$, namely M, should be numerously large to train a diffusion-based clone model. Hence, the above property of G_{def} is inherited by the clone model $\hat{G}$. Thus, we detect this property as a watermark in the detection phase (see Fig. 1).

C. Marking Phase in Defense Procedure

In the marking phase, we need to properly control G_{def}'s probability of generating marking images. This is because regular users do not need the marking images; thus, a high generation probability of marking images degrades G_{def}'s usability. In general, we can reduce the generation probability of marking images by decreasing the ratio of the marking dataset size $|\widetilde{\mathcal{X}}|$ to the main dataset size $|\mathcal{X}|$. However, the actual generation probability is not necessarily equal to $|\widetilde{\mathcal{X}}|/(|\mathcal{X}| + |\widetilde{\mathcal{X}}|)$. This is due to different loss scales between the main dataset and the marking dataset. Since images in the marking dataset have different characteristics (complexity, diversity, etc.) from those in the main dataset, their scales of loss calculated in Formula (2) are also different. If the loss for the marking dataset, $\mathcal{L}_{\mathrm{mark}}$, is larger than that for the main dataset, $\mathcal{L}_{\mathrm{main}}$, the $\mathcal{L}_{\mathrm{mark}}$ is preferentially reduced over $\mathcal{L}_{\mathrm{main}}$. As a result, marking images can be generated with a higher probability than expected.

To address the above issue, we extend the loss function of Formula (2) by introducing a weight w that balances between $\mathcal{L}_{\mathrm{main}}$ and $\mathcal{L}_{\mathrm{mark}}$. Specifically, we use

$$\begin{aligned} \mathcal{L}_{\mathrm{def}} &= \mathcal{L}_{\mathrm{main}} + w\mathcal{L}_{\mathrm{mark}} \\ &= ||\epsilon_t - \epsilon_\theta(t, x_{i,t})||^2 + w||\tilde{\epsilon}_t - \tilde{\epsilon}_\theta(t, \tilde{x}_{j,t})||^2 \end{aligned} \tag{3}$$

as a loss function for G_{def}. In this formula, $x_{i,t}$ and $\tilde{x}_{j,t}$ are a noised version of x_i and $\tilde{x}_j$, respectively, and ϵ_t and $\tilde{\epsilon}_t$ is a noise added to x_i and $\tilde{x}_j$, respectively. The weight w is set as $w = 1$ before the e-th epoch of the training while set as $w = \frac{\mathcal{L}_e}{\widetilde{\mathcal{L}}_e}$ after $(e+1)$-th epoch, where $\mathcal{L}_e$ and $\widetilde{\mathcal{L}}_e$ are the value of $\mathcal{L}_{\mathrm{main}}$ and $\mathcal{L}_{\mathrm{mark}}$ at the e-th epoch.

D. Detection Phase in Defense Procedure

In the detection phase, we judge whether a given suspicious model G' is a clone model of G_{def} or not. If G' is truly a clone model, it has a watermark, i.e., the property of generating marking images. In other words, G' can reconstruct a marking image from a noise image if and only if it is a clone model. To examine this point, we create a noise image $\tilde{x}_{j,t}$ from an original marking image $\tilde{x}_j \in \widetilde{\mathcal{X}}$ by applying the forward process of t steps and input the $\tilde{x}_{j,t}$ to the suspicious model G'. Let $\tilde{y}_j$ be the image reconstructed from $\tilde{x}_{j,t}$ by the reverse process of G'. We receive $\tilde{y}_j$ from G' and then examine whether it is a marking image or not.

To conduct the above examination, we compare $\tilde{y}_j$ and the marking dataset $\widetilde{\mathcal{X}}$ in the feature space. Specifically, we first extract a feature vector $\tilde{v}_j = E(\tilde{y}_j)$ from $\tilde{y}_j$, where E is the image encoder of CLIP [16]. At the same time, we also extract $v_j = E(\tilde{x}_j)$ from each original marking image $\tilde{x}_j \in \widetilde{\mathcal{X}}$ and average them, resulting in $v_{\mathrm{rep}} = \frac{1}{K}\sum_{j=1}^{K} v_j$. Then, we compute the cosine similarity between $\tilde{y}_j$ and v_{rep}, denoted by $s_j = \mathrm{sim}(v_{\mathrm{rep}}, \tilde{v}_j)$. If this similarity s_j exceeds a certain threshold $\tilde{\Theta}$, we can judge that $\tilde{y}_j$ has the same characteristics as $\widetilde{\mathcal{X}}$; that is, $\tilde{y}_j$ is a marking image. The threshold $\tilde{\Theta}$ is determined as follows, based on a statistical hypothesis. We hypothesize that the similarity $\mathrm{sim}(v_{\mathrm{rep}}, \tilde{v}_j)$

Fig. 2: Examples of images of datasets used in our experiments. [Left] CelebA, [Center] fruits-360, and [Right] church image dataset from LSUN.

follows a normal distribution, whose mean $\tilde{\mu}$ and variance $\tilde{\sigma}^2$ are estimated using $\widetilde{\mathcal{X}}$. Specifically, we calculate the cosine similarity $\mathrm{sim}(v_{\mathrm{rep}}, v_j)$ for all $j \in \{1, ..., K\}$ and obtain their mean and variance. We use these values as the estimation of $\tilde{\mu}$ and $\tilde{\sigma}^2$. Using them, we define the threshold $\tilde{\Theta}$ as

$$\tilde{\Theta} \;=\; \tilde{\mu} - 1.64\tilde{\sigma} \;, \tag{4}$$

which corresponds to a lower tail probability of 0.05 (5%) of a normal distribution.

To expand the clone detection capability of the proposed method, we conduct the above examination for all $\tilde{y}_j$. In other words, we calculate the similarity s_j for all $j \in \{1, ..., K\}$ and obtain a similarity set $\mathcal{S} = \{s_j = \mathrm{sim}(v_{\mathrm{rep}}, \tilde{v}_j) \,\big|\, j = 1, ..., K\}$. Then, we count how many s_j exceed the threshold of Formula (4). If G' is truly a clone model, many of s_j are expected to be larger than $\tilde{\Theta}$. Therefore, we compute the ratio of the number of s_j which exceeds $\tilde{\Theta}$ as

$$p_{\mathrm{mark}} \;=\; \frac{\left|\{s_j | s_j > \tilde{\Theta}\}\right|}{|\mathcal{S}|} \;. \tag{5}$$

This can be regarded as the success rate of reconstructing $\tilde{x}_j$ from $\tilde{x}_{j,t}$. If the success rate p_{mark} is high, G' is likely to be a clone model.

However, in some cases, p_{mark} could be high even if G' is not a clone model of G_{def}. For example, suppose the case that G_{def} is a face image generator trained with a marking dataset consisting of fruit images while G' is a fruit image generator independent from G_{def}. Such G_{def} has the capability of generating marking images even though it is not a clone. Therefore, p_{mark} becomes high. We should address this issue. To this end, we apply the above examination procedure not only to the marking dataset $\widetilde{\mathcal{X}}$ but also to the main dataset $\mathcal{X}$. More specifically, we attempt to reconstruct each $x_i \in \mathcal{X}$ from its noised version $x_{i,t}$ using G' and compute the success rate of this reconstruction process, denoted by p_{main}, in the same manner with p_{mark}. The p_{main} also becomes high if G' is a clone model, since it has almost the same generation capability as G_{def}. Hence, we judge that G' is a clone model if and only if both p_{mark} and p_{main} are higher than a certain threshold Φ. We will experimentally examine the clone detection accuracy of our proposed method with various Φ.

IV. Experiments on Attack Method

A. Experimental Setup

We experimentally evaluated the feasibility of MEA against DDIM using three datasets: (a) the face image dataset CelebA, (b) the fruit image dataset fruits-360, and (c) a church image dataset from LSUN. Fig. 2 shows examples of images of each dataset. For each of (a), (b), and (c), we randomly selected 16000 images and used them to train an original victim DDIM model G, which is denoted by DS hereafter. After training G, we also trained its clone model $\hat{G}$ by the procedure of Section III-A. The number of images for training $\hat{G}$ was also 16000. Note that all the images were resized to 128×128 pixels. Because the goal of adversaries is to create a clone model as similar as possible to the original victim model, we evaluated the FID [17] between images generated by G and those by $\hat{G}$. A lower FID indicates greater similarity between two generated image sets, namely between G and $\hat{G}$. In addition, we also evaluated the FID between G and DS as well as between $\hat{G}$ and DS. In the remainder, we describe the FID between A and B as $\mathrm{FID}(A, B)$. For example, $\mathrm{FID}(G, \hat{G})$ means the FID between the original model G and its clone model $\hat{G}$.

TABLE I: FID-based Evaluation of MEA on DDIM.

	DS		
	CelebA	fruits-360	LSUN church
$\mathrm{FID}(G, DS)$	21.14	13.60	21.38
$\mathrm{FID}(G, \hat{G})$	20.48	8.71	25.88
$\mathrm{FID}(\hat{G}, DS)$	51.32	38.22	64.35

TABLE II: Change in $\mathrm{FID}(G, \hat{G})$ with respect to the number of images used to train $\hat{G}$ in the case of CelebA.

Num. of training images for $\hat{G}$	8000	16000	32000
FID	22.77	20.48	16.53

B. Results

Table I presents the evaluation results of $\mathrm{FID}(G, \hat{G})$, $\mathrm{FID}(G, DS)$, and $\mathrm{FID}(\hat{G}, DS)$ for each dataset. As shown in this table, $\mathrm{FID}(G, \hat{G})$ is sufficiently small. In addition, Fig. 3 compares some images generated by G and those by $\hat{G}$, where $\hat{G}$ can generate images of almost the same quality as G. These results suggest the feasibility of MEA on DDIM; in other words, there is an unignorable risk of MEA on diffusion models. However, $\mathrm{FID}(\hat{G}, DS)$ are not comparably high than $\mathrm{FID}(G, DS)$. This is attributed not only to the difference in distributions of DS and G but also to that of G and $\hat{G}$. Due to these two differences multiplied by each other, there is a large difference between the distribution of DS and that of $\hat{G}$. In

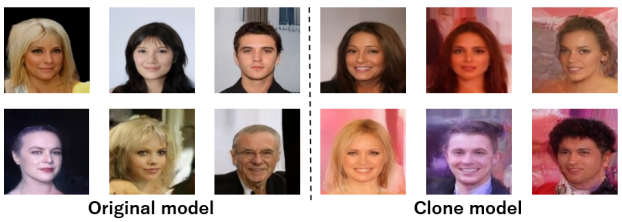

Fig. 3: Comparison of images generated by original victim model G and its clone model $\hat{G}$ in the case of CelebA. [Left] images generated by G. [Right] images generated by $\hat{G}$.

TABLE III: Effect of w in Formula (3) on the generation rate of marking images. The bolded values are closer than the non-bolded values to the ideal generation rate in this experiment, namely $|\widetilde{\mathcal{X}}|/\left(|\mathcal{X}|+|\widetilde{\mathcal{X}}|\right) = 3.03\%$.

Case	Training dataset of G_{def}		Generation rate of marking images	
	Main dataset	Marking dataset	with w	without w
(i)	CelebA	fruits-360	5.09%	**1.77%**
(ii)	CelebA	LSUN church	**2.33%**	2.27%
(iii)	fruits-360	CelebA	**2.41%**	5.93%
(iv)	fruits-360	LSUN church	**4.76%**	7.81%
(v)	LSUN church	CelebA	**3.08%**	5.37%
(vi)	LSUN church	fruits-360	**4.18%**	4.35%

other words, it also depends on the performance of the original model G whether its clone model $\hat{G}$ can generate plausible images or not.

Table II evaluates how much the performance of the clone model $\hat{G}$ is affected by the number of its training images, focusing on the case of CelebA. As shown in this table, $\hat{G}$ can achieve the FID of 22.77 even with only 8000 images received from the original victim model G. This indicates that adversaries can build a clone model with fewer query accesses than the amount of training data for the original model. This is a consistent result with existing work focusing on GAN and VAE [6], [8], demonstrating the risk of MEA.

V. Experiments on Defence Method

A. Experimental Setup

Next, we experimentally evaluated the effectiveness of our defense method. The datasets used in this experiment are the same as those used in the last section: (a) CelebA, (b) fruits-360, and (c) LSUN church. When training a defense model G_{def}, we selected one of them as the main dataset and another as the marking dataset. The size of the main dataset was set to 16000, while that of the marking dataset was set to 500. We attempted this with all combinations of (a), (b), and (c), creating six defense models. After that, for each of the six models, we trained its clone model $\hat{G}$. In addition, we also trained a normal non-clone model using only the main dataset, i.e., a subset of either (a), (b), or (c) containing 16000 images. We repeated this twice with different subsets, creating six non-clone models. For the marking method, we evaluated the impact of the balancing weight w introduced in Formula (3). To this end, we calculated the defense model's generation ratio of marking images in the case with w and that without w and compared their results. For the detection method, we evaluated its False Acceptance Rate (FAR) and False Rejection Rate (FRR) using the above 12 models. More specifically, we applied our detection method to both the clone models and the non-clone models and classified each of them based on p_{mark} and p_{main}. We varied the threshold Φ from 0 to 1 and observed the change in FAR and FRR.

B. Results on Marking Performance

Table III shows the generation rate of marking images for each defense model. Since we used 16000 images as a main dataset $\mathcal{X}$ and 500 images as a marking dataset $\widetilde{\mathcal{X}}$, the ideal rate expected from the size of the main and marking datasets is $\frac{|\widetilde{\mathcal{X}}|}{|\mathcal{X}|+|\widetilde{\mathcal{X}}|} = \frac{500}{16000+500} = 3.03\%$. As seen in this table, introducing w effectively makes the marking image generation rate closer to the ideal value and maintains the usability of G_{def} more.

TABLE IV: Evaluation of the clone models' p_{mark} and p_{main}

Case	Training dataset of G_{def}		Reconstruction rate of marking images by $\hat{G}$	
	Main dataset	Marking dataset	p_{mark}	p_{main}
(i)	CelebA	fruits-360	0.982	0.930
(ii)	CelebA	LSUN church	0.676	0.992
(iii)	fruits-360	CelebA	0.948	0.994
(iv)	fruits-360	LSUN church	0.884	0.948
(v)	LSUN church	CelebA	0.962	0.948
(vi)	LSUN church	fruits-360	0.956	0.900

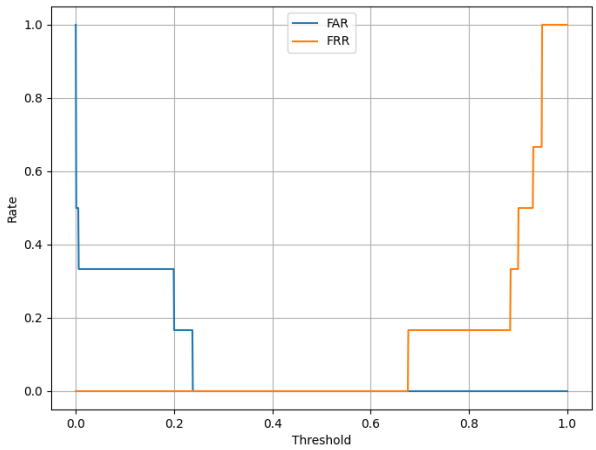

Fig. 4: Change in FAR and FRR for clone model detection with various Φ. FAR is the error rate of misclassifying normal (or non-clone) models as clone models. FRR is the error rate of misclassifying clone models as normal (or non-clone) models.

C. Results on Detection Performance

Fig. 4 shows the result of clone model detection. As seen in this figure, we can achieve $\mathrm{FAR} = \mathrm{FRR} = 0\%$, namely a detection rate of 100%, by setting the threshold Φ between 0.238 and 0.676. This is because both p_{mark} and p_{main} are lower than 0.238 for all non-clone models while higher than 0.676 for all clone models. This large difference indicates the effectiveness of our detection method.

Table IV shows the values of p_{mark} and p_{main} for each of the six clone models. In this table, p_{main} is very high (more than 0.9) for all the clone models. In contrast, p_{mark} is not always very high; it is higher than 0.94 when CelebA and fruits-360 were used as the marking dataset to train G_{def}, whereas lower than 0.89 when LSUN church was used as the marking dataset. This would be due to the difference in the diversity of each image set. The high diversity of a marking dataset makes its distribution less peaked. Such a less-peaked distribution is difficult to precisely clone since the number of marking images in the training dataset of clone models is small. As a result,

TABLE V: Quality comparison of images generated by defense models and those by normal models in terms of FID. We computed FID between the generated images and the main dataset, excluding any marking image from this evaluation.

(a) Case of CelebA as main dataset

Model	Marking dataset	FID↓
Defense model	fruits-360	25.31
	LSUN church	19.66
Normal model (without defense)	–	21.14

(b) Case of fruits-360 as main dataset

Model	Marking dataset	FID↓
Defense model	CelebA	22.03
	LSUN church	30.05
Normal model (without defense)	–	13.60

(c) Case of LSUN church as main dataset

Model	Marking dataset	FID↓
Defense model	CelebA	24.66
	fruits-360	24.60
Normal model (without defense)	–	21.38

p_{mark} becomes relatively low. As shown in Fig. 2, all the images in fruits-360 share a similar structure, a fruit centered on a white background, making the distribution of marking images more peaked. In contrast, images in LSUN church have highly diverse structures. For instance, some images have an internal appearance of a church, while others have an outside appearance. Therefore, the distribution of marking images becomes less peaked. Although the values of p_{mark} in Table IV are sufficiently high to achieve a good detection result in this experiment, we conclude that a less diverse marking dataset is preferable to achieve more reliable clone model detection.

D. Evaluation of Defense Model's Image Quality

The marking images added to train a defense model might degrade the quality of its generated images. To examine the seriousness of this effect, we compared the image quality of the defense models and that of the undefended normal models in terms of FID. Table V shows the result. Note that we excluded the marking images generated by the defense models from the calculation of FID. We can see from Table V that all the defense models achieve an FID of around 30 or less, showing no serious issue in the image quality of the defense models. Compared to the undefended models, the defense models achieve a comparable quality when CelebA and LSUN church are used as the main dataset. However, in the case of using fruits-360 as the main dataset, the defense models underperform the undefended models by 10-15 points. We consider that this quality degradation is due to the relatively high diversity of the marking dataset, i.e., CelebA and LSUN church, compared to the main dataset, i.e., fruits-360. This finding suggests the importance of carefully considering the trade-off between the detection performance and the risk of image quality degradation.

VI. Conclusion

In this paper, we first experimentally demonstrated the feasibility of MEA on DDIM and then proposed its countermeasure, namely, a defense method for detecting clone models created by MEA. To achieve this, the proposed method embeds a property of occasionally generating OOD images, called "marking images", into the original victim model as a watermark. This property is inherited by cloned models, allowing us to detect them based on the presence or absence of the watermark. In the results of our experiments, our method successfully detected all the clone models without seriously degrading the usability of the original model and the quality of its generated images. In our future work, we will develop a more sophisticated approach for constructing a marking image set to boost the robustness of the proposed method. Making marking images less detectable by human eyes (i.e., the adversary's eyes) is another important future work. This work was supported by JSPS KAKENHI Grant Number JP25K03121.

References

[1] F. Tramer, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *Proc. the 25th USENIX Security Symposium*, 2016, pp. 601–618.

[2] Y. Shi, Y. Sagduyu, and A. Grushin, "How to steal a machine learning classifier with deep learning," in *Proc. IEEE Int'l Symposium on Technologies for Homeland Security*, 2018, pp. 1–5.

[3] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning," in *Proc. the IEEE Symposium on Security and Privacy*, 2018, pp. 36–52.

[4] Oh, S. Joon, Schiele, Bernt, Fritz, and Mario, *Towards Reverse-Engineering Black-Box Neural Networks*. Springer Int'l Publishing, 2019.

[5] T. Orekondy, B. Schiele, and M. Fritz, "Prediction poisoning: Towards defenses against dnn model stealing attacks," in *Proc. the 8th Int'l Conf. on Learning Representations*, 2020, pp. 1–17.

[6] H. Hu and J. Pang, "Stealing machine learning models: Attacks and countermeasures for generative adversarial networks," in *Proc. the 37th Annual Computer Security Applications Conf.*, 2021, pp. 1–16.

[7] I. J. Goodfellow, J. Pouget-Abadie, B. X. M. Mirza, D. Warde-Farley, A. C. S. Ozair, and Y. Bengio, "Generative adversarial nets," in *Proc. the 27th Int'l Conf. on Neural Information Processing Systems*, vol. 2, no. 9, 2014, pp. 2672–2680.

[8] S. Liu, "Model extraction attack and defense on deep generative models," *Conf. Series*, vol. 2189, no. 1, pp. 2672–2680, 2022.

[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. the 2nd Int'l Conf. on Learning Representations*, 2014, pp. 1–14.

[10] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. The 33rd Int'l Conf. on Machine Learnings*, 2016, pp. 1747–1756.

[11] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," in *Proc. the 34th Int'l Conf. on Neural Information Processing Systems*, 2021, pp. 8780–8794.

[12] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. the 9th Int'l Conf. on Learning Representaions*, 2021, pp. 1–22.

[13] E. L. Merrer, P. Perez, and G. Tredan, "Adversarial frontier stitching for remote neural network watermarking," *Neural Computing and Applications*, vol. 32, no. 13, pp. 9233–9244, 2020.

[14] J. Tan, N. Zhong, Z. Qian, and X. Z. S. Li, "Deep neural network watermarking against model extraction attack," in *Proc. the 31st ACM Int'l Conf. on Multimedia*, 2023, pp. 1588–1597.

[15] P. Dhariwal and A. Nichol, "Denoising diffusion probabilistic models," in *Proc. the 34th Int'l Conf. on Neural Information Processing Systams*, 2020, pp. 6840–6851.

[16] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proc. the 38th Int'l Conf. on Machine Learning*, 2021, pp. 8748–8763.

[17] 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," in *Proc. Int'l Conf. on Neural Information Processing Systems*, 2017, pp. 6629–6640.