Fingerprinting Denoising Diffusion Probabilistic Models

Huan Teng[1,2] Yuhui Quan[1] Chengyu Wang[2] Jun Huang[2] Hui Ji[3]

[1]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

[2]Alibaba Cloud Computing, Hangzhou 311121, China

[3]Department of Mathematics, National University of Singapore, 119076, Singapore

huan.teng.cs@foxmail.com, csyhquan@scut.edu.cn, {chengyu.wcy, huangjun.hj}@alibaba-inc.com, matjh@nus.edu.sg

Abstract

Diffusion models, especially denoising diffusion probabilistic models (DDPMs), are prevalent tools in generative AI, making their intellectual property (IP) protection increasingly important. Most existing IP protection methods for DDPMs are invasive, e.g.*, model watermarking, which alter model parameters and raise concerns about performance degradation, also with requirement for extra computational resources for retraining or fine-tuning. In this paper, we propose the first non-invasive fingerprinting scheme for DDPMs, requiring no parameter changes or fine-tuning, and keeping generation quality intact. We introduce a discriminative and robust fingerprint latent space based on the well-designed "crossing route" of noisy samples that span the performance border-zone of DDPMs, with only black-box access required for the diffusion denoiser in ownership verification. Extensive experiments demonstrate that our fingerprinting approach enjoys both robustness against the often-seen attacks and distinctiveness on various DDPMs, providing an alternative for protecting DDPMs' IP rights without compromising their performance or integrity*[1].

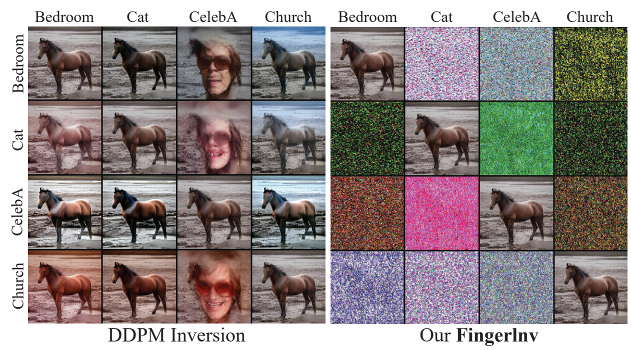

Figure 1. Cross-generation results for DDPM inversion [10] and **FingerInv**. In each confusion matrix, we invert the same "horse" image for different DDPMs to obtain their latent codes, which are then used to generate image for different DDPMs. Each row uses a specific latent code across various DDPMs, and the diagonal positions indicates matched DDPMs and latent codes. DDPM inversion shows universality but lacks distinctiveness, while **FingerInv** only reconstructs matched cases, highlighting distinctiveness.

1. Introduction

Diffusion models (DMs) [29, 31], particularly DDPMs [8], have become essential models in artificial intelligence generated content (AIGC), renowned for their exceptional generative capabilities and wide-ranging applications [3, 14, 26, 32]. With their increasing adoption, especially through open-source platforms, the need to protect their IP rights and guard against misuse has become critical. While open-source models foster rapid innovation, they also introduce vulnerabilities, allowing unauthorized exploitation for illicit purposes, including IP infringement [11, 18, 33].

The current leading IP protection solutions for DDPMs are invasive, such as model watermarking [4, 16, 22, 40, 44], which embeds specific information (called watermark) into the model by modifying its parameters for ownership verification. However, watermarking affects model performance and adds computational overhead during training or fine-tuning. Watermarking methods are divided into black-box [1, 15, 15, 24, 41] and white-box schemes [20, 27, 34], with black-box approaches preferred for their lower verification requirements. Nonetheless, both schemes inherently modify the model and can be resource-intensive.

1.1. Motivation

Recently, model fingerprinting [2, 6, 17, 23, 25, 43] has gained significant attention as a non-invasive approach to protecting neural network (NN) models in image classification and restoration. Model fingerprinting involves calculating a unique identifier within a model or its behavior, enabling ownership verification without altering its performance, and requiring no additional training or fine-tuning.

[1]Source codes are released in https://github.com/painfulloop/Fingerprint_DDPM

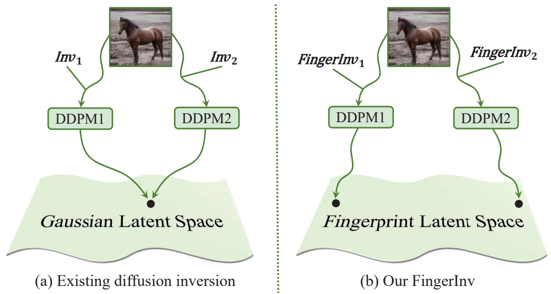

Figure 2. Illustration of the basic idea for our inversion. We aim to define a discriminative fingerprint latent space using **FingerInv**.

To the best of our knowledge, no existing work has studied fingerprinting for DDPMs. Current fingerprinting methods are tailored for deterministic NNs that map an input image to a label via one-to-one mapping. In contrast, DDPMs are probabilistic models that map a standard distribution to an image distribution, allowing for generation of new images from random samples (*e.g.*, Gaussian noise). Due to such a fundamental difference between probabilistic DDPMs and deterministic NNs for classification or restoration, existing fingerprinting methods cannot be easily adapted to DDPMs. This motivated us to develop a new fingerprinting method specifically designed for DDPMs.

1.2. Main Idea

Consider a diffusion generation process $\mathcal{G}(z) = x_0$, where z is the latent code and $x_0 \in \mathbb{R}^{H\times W\times C}$ is the resulting image. Our fingerprinting method, called **FingerInv** and denoted as $\mathcal{F}$, uses a predefined verification image x_0 containing copyright details, and inverts $\mathcal{G}$ to find a distinctive fingerprint latent code $z = \mathcal{F}(x_0, \mathcal{G})$, serving as the trigger key. During verification, z is input into the suspect model, and the output is validated for ownership, requiring only black-box access of the DDPM denoiser $\epsilon_{\boldsymbol{\theta}}$. The key concern is defining $\mathcal{F}$ to ensure the distinctiveness of the latent code z. It has been observed that, when utilizing a fixed "random seed", two DMs tend to produce similar images [21, 38]. Furthermore, denoisers trained on non-overlapping datasets can potentially learn nearly identical score functions [12]. Thus, existing diffusion inversion methods [10, 38] may exhibit similar Gaussian latent spaces. As shown in Figure 1 (left), applying existing inversion methods on Gaussian noise space [10] results in interchangeable latent codes z. Most models can reconstruct x_0 using latent codes of other models, lacking uniqueness.

We propose to utilize our **FingerInv** to map the verification images to the discriminative fingerprint latent space as illustrated in Figure 2. Our fingerprint latent code can be used directly for white-box verification and also enhance discriminability for black-box verification, as shown in Figure 1. In DDPM, the denoiser $\epsilon_{\boldsymbol{\theta}}$ is typically trained to estimate noise ϵ_t within the Gaussian noisy domain $\mathbb{D}$, resulting in small prediction errors $\|\epsilon_{\boldsymbol{\theta}}(x_t) - \epsilon_t\|_2^2$ for samples from $\mathbb{D}$. Besides, there is also a complementary set $\bar{\mathbb{D}}$ that causes $\epsilon_{\boldsymbol{\theta}}$ to produce large prediction errors, and the boundary region between $\mathbb{D}$ and $\bar{\mathbb{D}}$ is defined as the performance border-zone, which possesses good distinctiveness and robustness [25]. Inspired by adversarial samples across the decision boundary, leveraging the characteristic of DDPM generation through progressive denoising, we propose the discriminative "crossing route" on performance border-zone to construct unique noisy samples.

As illustrated in Figure 3, the crossing route is defined as a set of noisy samples $\{x_1, \ldots, x_T\}$ that precisely span the performance border-zone. After obtaining these samples, we can back-derive other latent components [10, 38] based on DDPM sampling process to obtain the entire fingerprint latent code $z = \{x_T, z_1, \ldots, z_T\}$. Moreover, compared to the critical point [25] in performance border-zone, our crossing route includes noisy samples with sufficient large prediction errors (*e.g.*, x_T in $\bar{\mathbb{D}}$). These "difficult" samples often have potential to enhance the distinction in model output domain, which is crucial for black-box verification.

1.3. Contribution

We utilize **FingerInv** to obtain the fingerprint latent code as the trigger key from the verification image, and create a trigger-verification pair for IP protection. Our method is primarily validated on two representative approaches: pixel space DDPMs (PS-DDPMs) [8], and latent diffusion models (LDMs) [26]. Extensive experiments show that our proposed method exhibits greater distinctiveness and robustness compared to baseline fingerprint methods , while remaining competitive against recent invasive watermarking techniques. The main contributions are listed below:

- We propose the first non-invasive fingerprinting framework aimed at protecting IP rights for DDPMs. The verification process is simple and intuitive, requiring only black-box access to the DDPM denoiser, without additional visualization components.
- Inspired by adversarial samples across decision boundaries in classification, combining with DDPM iterative scheduling, the concept of crossing route on performance border-zone is introduced to characterize DDPMs.
- A distinctive and robust fingerprint latent space is proposed. By mapping the verification image to the fingerprint latent code, the model owner can obtain the trigger-verification pair, serving to protect IP right.

2. Related Work

2.1. Invasive Methods

Most watermarking approaches for DDPMs are invasive [4, 9, 16, 22, 40, 44]. [44] proposed watermarking strategies

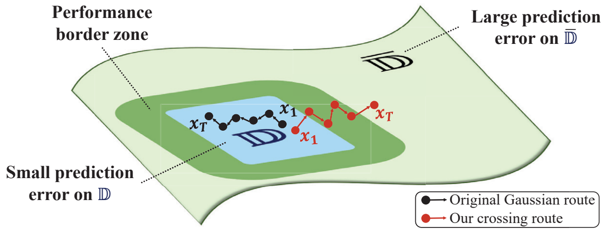

Figure 3. Main concept of our crossing route. We identify a distinct path $\{x_1, \ldots, x_T\}$ that crosses the performance boundary of $\epsilon_{\boldsymbol{\theta}}$. Assuming x_0 in-domain, unlike original Gaussian route within $\mathbb{D}$, ours transitions from $\mathbb{D}$ to $\bar{\mathbb{D}}$ during the diffusion process.

for both unconditional and conditional DMs, which introduced to watermark the data before training the models for unconditional generation, and fine-tune the DMs to embed a special trigger prompt and predefined verification images (*e.g.*, QR codes) for text-to-image generation. In addition to text prompts [16, 44], backdoor watermarking in DDPMs can also use image triggers [22]. To improve fidelity, a two-stage approach is proposed to separately fine-tune the text encoder and U-Net for watermarking stable diffusion (SD) models [40]. Stable Signature invasively fine-tunes the LDM decoder to embed watermark information for all generated images [4] (rather than the specific output of a trigger), while fine-tuning the decoder again with unwatermarked samples can erase the watermark [9]. In conclusion, watermarking DDPM or its outputs always requires training or fine-tuning of models, which incurs resource costs and potentially alters model performance.

2.2. Non-invasive Methods

In image classification, a commonly used non-invasive method for model copyright protection is fingerprinting [2, 6, 17, 23, 43], which typically identifies samples on decision boundaries or adversarial samples and distinguishes models based on their varying behaviors. The fingerprinting method also exists in image restoration [25], which finds critical points within performance border- zone, demonstrating that the critical points of various image restoration models exhibit good distinctiveness and robustness. However, it requires white-box access in verification stage, thus posing practical challenges in real-world applications. In DDPMs, an analogous fingerprinting approach [37] primarily protects the copyright of generated images rather than the IP rights of the model itself. Furthermore, [37] watermarks the original latent space of DMs, which affects the quality of the generated samples, making it invasive.

2.3. Inversion Methods for DDPMs

Model inversion is typically applied in fingerprinting classification and restoration models [2, 5, 25]. Given outputs, when it comes to inversely acquiring the latent codes for DMs, several existing methods have been proposed [10, 19, 30, 35, 38]. For deterministic sampling process, these include the DDIM inversion [30], as well as null-text inversion [19], utilizing DDIM inversion as pivot and optimizing null-text embeddings, and EDICT [35], the inversion approach via coupled transformations. For stochastic sampling process, CycleDiffusion [38] recovers the sequence of noise vectors to perfectly reconstruct the image, and a more edit-friendly variant of CycleDiffusion, DDPM inversion [10] is proposed. Compared with the deterministic inversion methods, the stochastic ones have higher-dimensional latent space, which makes it easier to obtain better uniqueness and robustness in our fingerprinting task. Thus, after searching for crossing routes on the performance border-zone, our method obtains the fingerprint latent code based on the spirit of stochastic inversion methods [10, 38].

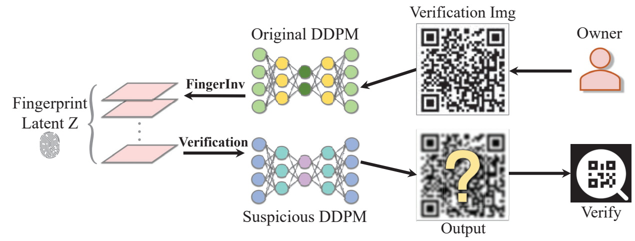

Figure 4. Framework of our fingerprinting approach.

3. Methodology

3.1. Problem Statement and Overview

Threat model Similar to [25], in common scenarios of IP protection, the model owner trains the model using their private resources, while an attacker attempts to steal the model. The model owner, acting as the defender, needs to have the ability to verify whether a suspicious model is a plagiarized version. Meanwhile, the attacker needs to modify the model to evade detection of ownership while ensuring that the modified model retains its functionality and performance. Typically, the modifications may involve common techniques such as pruning, quantization, and fine-tuning. Moreover, unlike [25], our ownership verification imposes stricter conditions by allowing only black-box access to the denoiser, without access to gradient information.

Principles Model fingerprinting typically considers two requirements [2, 25]: first, discriminability/uniqueness, which mandates that different models possess unique fingerprints and ensures that the fingerprints of other models do not trigger a specific model's verification information, and vice versa; second, robustness, which requires that the fingerprint can still successfully trigger the verification message even after the model has been attacked or modified.

Framework Figure 4 illustrates our framework:

- First, select a verification image x_0 that contains the copyright information as the target output.
- Next, perform **FingerInv** to map the target verification image x_0 to fingerprint latent code (trigger) z.
- Then, to verify a suspicious DDPM, using z to generate an image and check if it contains copyright information.

Inversion choice To define **FingerInv**, the reconstruction capability for the given verification image x_0 is crucial. In diffusion inversion schemes, DDIM inversion addressing the deterministic sampling process are based on linearization assumption [35], leading to error accumulation. Although some approaches mitigate this error, they often incur additional overhead, such as optimization [19] or doubling the sampling computational costs [35]. In contrast, Cycle Diffusion [38] and DDPM inversion [10] ensure perfect reconstruction for a given image, and compared to methods that use only x_T as the latent code, these methods leverage more information $\{x_T, z_T, \ldots, z_1\}$, indicating a greater potential to enhance discriminability and robustness.

3.2. Fingerprint Extraction

Preliminaries DDPM adds noises to the clean image x_0 during the forward process to gradually obtain white Gaussian noise x_T and reverses this process during sampling. The forward process can be expressed as follows:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_t, \tag{1}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, α_s denotes a specified variance schedule, and $\epsilon_t \sim \mathcal{N}(0, \mathbf{I})$. This process is commonly employed to construct a posterior distribution $q(x_{1:T}|x_0)$ to obtain the noisy images from x_1 to x_T. Subsequently, the DDPM sampling process is defined by:

$$x_{t-1} = \hat{\mu}_t(x_t) + \sigma_t z_t, \quad t = T, \ldots, 1, \tag{2}$$

where z_t are sampled i.i.d. standard Gaussian noises and the mean estimator $\hat{\mu}_t(x_t)$ is defined as:

$$\hat{\mu}_t(x_t) = \sqrt{\bar{\alpha}_{t-1}} \hat{x_0} + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \epsilon_{\boldsymbol{\theta}}(x_t), \tag{3}$$

where $\hat{x_0} = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_{\boldsymbol{\theta}}(x_t)}{\sqrt{\bar{\alpha}_t}}$ is predicting x_0, and the second term represents the direction pointing to x_t; $\epsilon_{\boldsymbol{\theta}}$ represents the denoiser and the variance schedule σ_t is defined as $\eta \frac{\beta_t(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)}$, where $\eta \in [0, 1]$. Specifically, $\eta = 1$ corresponds to the DDPM and $\eta = 0$ to the DDIM scheme. CycleDiffusion and DDPM inversion utilize Eq. (1) to add dependent/independent Gaussian noise to obtain a set of noisy images $\{x_1, \ldots, x_T\}$. Subsequently, the inversion process entails back-calculating $\{z_T, \ldots, z_1\}$ based on Eq. (2) to ensure perfect reconstruction. Thus, the latent components z_t can be simply inferred using:

$$z_t = \frac{x_{t-1} - \hat{\mu}_t(x_t)}{\sigma_t}, \quad t = T, \ldots, 1. \tag{4}$$

Initialization As discussed in Section 1.2, we seek to leverage the crossing route through the performance border-zone of $\epsilon_{\boldsymbol{\theta}}$ to implement our **FingerInv**. According to Eq. (1), we can change the distribution of Gaussian noise ϵ_t to achieve it, denoted as $\tilde{\epsilon_t}$. With the noises $\{\tilde{\epsilon_1}, \ldots, \tilde{\epsilon_T}\}$, our goal is to make the noisy samples $\{x_1, \ldots, x_T\}$ precisely traverse through the performance border-zone of the DDPMs. We define the initialized $\tilde{\epsilon}_t^{(0)}$ as follows:

$$\tilde{\epsilon}_t^{(0)} = \delta_1 \frac{t-1}{T} n_o + n_g, \tag{5}$$

where $n_g \sim \mathcal{N}(0, \mathbf{I})$, n_o is from a non-Gaussian distribution, and δ_1 is a weight controlling the initial intensity; *e.g.*, n_o could be uniformly distributed: $n_o \sim \mathcal{U}(-1, 1)$.

According to Eq. (5), when $t = 1$, $\tilde{\epsilon}_t^{(0)} = n_g$, indicating that $x_1^{(0)}$ is easy to predict for the DDPM denoiser $\epsilon_{\boldsymbol{\theta}}$. As t increases, the intensity of n_o also increases, resulting in x_t becoming more disordered and further deviating from the original Gaussian domain, which implies that $x_t^{(0)}$ becomes increasingly difficult for $\epsilon_{\boldsymbol{\theta}}$ to predict. We try to make that, during the initialization phase, $\{x_1^{(0)}, \ldots, x_T^{(0)}\}$ traverse the performance border-zone of the DDPM as possible.

Optimization To ensure that noisy samples reflect the inherent capabilities of $\epsilon_{\boldsymbol{\theta}}$ and serve as the crossing route, we optimize the noise $\tilde{\epsilon}_t$ with $\epsilon_{\boldsymbol{\theta}}$ fixed. The loss function is:

$$L_{\text{critical}} = \frac{T-t}{T} \|\epsilon_{\boldsymbol{\theta}}(x_t) - \tilde{\epsilon}_t\|_2^2 - \delta_2 \frac{t-1}{T} \|\nabla x_t\|_1, \tag{6}$$

where δ_2 is the weight parameter. The first term supports the denoiser in predicting the noise in x_t, whereas the second term increases the total variation (TV, the ℓ_1-norm of image gradient) of x_t, making noise prediction more challenging, as discussed in [25]. While [25] hypothesizes that the critical point in performance border-zone has good uniqueness and confirms its white-box performance, it suffered in the black-box situation. The problem may be that [25] fixed the artificial degradation process and optimizes the clean image, making recovery easy in samples with small degradation. So we add TV loss on noisy samples rather than clean samples, and by fixing x_0, it is equivalent to directly optimizing the noise to make restoration hard, which potentially increases black-box discrimination.

Eq. (6) positions x_t to the performance border-zone of $\epsilon_{\boldsymbol{\theta}}$. As shown in Figure 3, when $t = 1$, only the first term is used, ensuring that the noise of x_1 is easily predicted by $\epsilon_{\boldsymbol{\theta}}$, placing it within $\mathbb{D}$. When $t = T$, only the second term is active, ensuring that the noise of x_T is difficult to predict by $\epsilon_{\boldsymbol{\theta}}$, placing it within $\bar{\mathbb{D}}$. Therefore, we obtain the crossing route $\{x_1, \ldots, x_T\}$ that traverse the performance border-zone, possessing sufficient discriminative properties.

Thus, calculating the latent code z via Eq. (4) becomes more discriminative.**FingerInv** is detailed in Algorithm 1.

Algorithm 1 Fingerprint Inversion Algorithm

Input: $\epsilon_{\boldsymbol{\theta}}$: DDPM denoiser, x_0: verification image, T: number of timesteps, δ_1 and δ_2: hardness parameters, λ: learning rate, N: optimization steps in each timestep

Output: latent code $z = \{x_T, z_T, \dots, z_1\}$

for $t = 1$ **to** T **do** // Stage1: obtain $\{x_1, \dots, x_T\}$
  // Noisy samples across performance border-zone
  $n_o \sim \mathcal{U}(-1, 1), n_g \sim \mathcal{N}(0, 1)$
  $\tilde{\epsilon}_t = n_g + \delta_1 \frac{t-1}{T} n_o$
  for $i = 1$ **to** N **do** // Optimize $\tilde{\epsilon}_t$
    $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \tilde{\epsilon}_t$
    $L_{\text{critical}} = \frac{T-t}{T} \|\epsilon_{\boldsymbol{\theta}}(x_t) - \tilde{\epsilon}_t\|_2^2 - \delta_2 \frac{t-1}{T} \|\nabla x_t\|_1$
    // Update $\tilde{\epsilon}_t$ using gradient descent
    $\tilde{\epsilon}_t = \tilde{\epsilon}_t - \lambda \nabla_{\tilde{\epsilon}_t} L_{\text{critical}}$
  end for
  $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \tilde{\epsilon}_t$
end for
for $t = T$ **to** 1 **do** // Stage2: obtain $\{z_T, \dots, z_1\}$
  $z_t \leftarrow (x_{t-1} - \hat{\mu}_t(x_t))/\sigma_t$
end for
return latent code $z = \{x_T, z_T, \dots, z_1\}$

Selection of the verification image We propose using QR codes as verification images for two reasons: ease of verification, and that QR images are typically not within the target domain of DDPMs, which are more likely to achieve better distinctiveness and robustness [24]. Note that although QR codes are a natural choice for verification due to their scan-based validation and are widely used in e-commerce, our method is not restricted to QR codes. Some results using natural images as verification images are provided in Section 7.2 of the supplementary material.

4. Experiment

4.1. Experimental Setup

Source models For PS-DDPMs, we explored classic DDPMs [8] on the LSUN [39] and CelebA-HQ [13] datasets, focusing on church, cat, bedroom, and face images. We used four generative models at 256x256 resolution with exponential moving average (EMA) techniques. For LDMs, we used models like SD V1-4 [26], Pixart-α [3], and the float16 DeciDiffusion [32]. All pretrained models are accessible online.[2] PS-DDPMs share an architecture but differ in datasets, while LDMs are based on the similar LAION [28] datasets with different structures. This variety

[2] https://huggingface.co/google
https://huggingface.co/Deci/DeciDiffusion-v1-0
https://huggingface.co/PixArt-alpha/PixArt-XL-2-512x512
https://huggingface.co/CompVis/stable-diffusion-v1-4

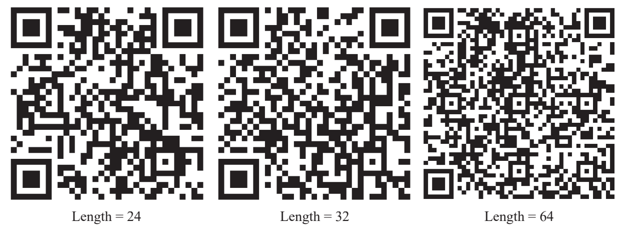

Figure 5. QR code images used in our experiments.

enhances the reliability and validity of our discriminability experiments, all conducted on an H800 GPU.

Implementation details In Algorithm 1, we set $\delta_1 = 20$; $\delta_2 = 1$; $N = 10$; $\lambda = 0.1$. As mentioned in Section 3.2, we use the QR code images as the verification image. Let l_{qr} denote the length of the string encoded in the QR code image. We set l_{qr} to 24, 32, and 64 to investigate the impact of different l_{qr} on fingerprinting, and use random strings to generate QR codes. As shown in Figure 5, the complexity of QR code patterns increases with l_{qr}. For LDMs, we utilized 512x512 resolution QR codes, while for PS-DDPMs, we downsampled the QR codes to 256x256 to facilitate comparison. We found that l_{qr} has minimal impact on the results, thus primarily present verification results for $l_{qr} = 32$ in part of our subsequent analyses. Results for more l_{qr} are given in Sections 9 and 10 of the supplementary material.

Baselines As we are the first to develop a non-invasive method specifically for DDPMs, we considered employing existing inversion techniques for DDPMs as baselines for comparison, including $\text{Cycle}_{\text{Diff}}$ (Cycle Diffusion [38]) and DDPM_{inv} (DDPM inversion [10]). In addition, we also incorporated several existing watermarking schemes for comparison, including invasive methods such as those presented in WM_{DM} (WatermarkDM [44] for PS-DDPMs and LDMs) and $\text{Stable}_{\text{sig}}$ (Stable Signature [4] for LDMs, which aim to protect generated images and can also reflect model IP) .

4.2. Uniqueness Analysis

Quantitative analysis in fingerprint domain We compared the distinctiveness of fingerprints by obtaining latent codes z using $\text{Cycle}_{\text{Diff}}$ (original Gaussian space), DDPM_{inv} (edit-friendly space), and our **FingerInv**. We calculated the distances between latent codes for PS-DDPMs and LDMs, averaging over different lengths l_{qr} for comparison.

Specifically, we tested the squared l_2 distance between an owner fingerprint z and a suspicious fingerprint z', with defined thresholds $\gamma_{ps} = 8.27 \times 10^6$ for PS-DDPMs and $\gamma_{ldm} = 6.91 \times 10^5$ for LDMs as per [24, 25] (see Section 6.2 of supplementary material). A distance below the threshold indicates theft; above, no theft. Figure 6 shows that our method produces distances below thresholds along the diagonal only, effectively distinguishing mod-

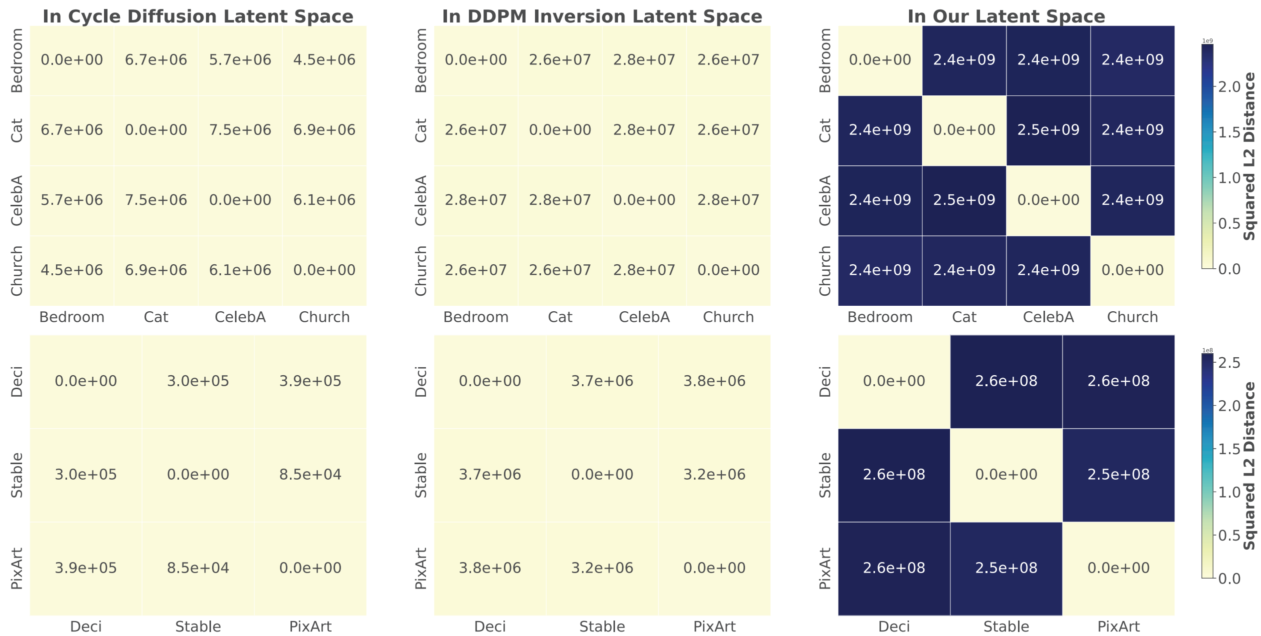

Figure 6. Squared l_2 distances in different latent spaces. The top row of the confusion matrices is for PS-DDPMs, while the bottom row is for LDMs. Columns show the results of CycleDiffusion, DDPM inversion, and **FingerInv**. Yellow hues indicate higher similarity between latent codes, whereas blue signifies greater dissimilarity. Our fingerprint latent space exhibits significantly better discriminability.

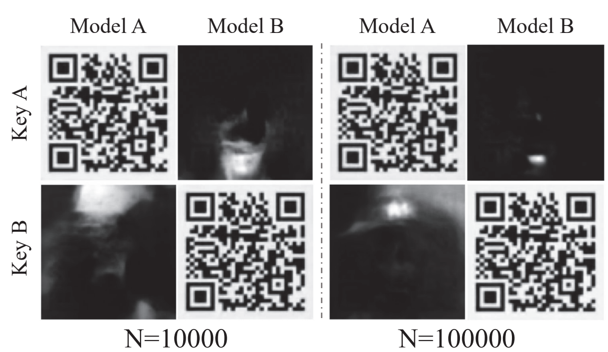

Figure 7. Discriminability analysis for highly similar denoising generative models with nearly the same score function and density.

els. $\text{Cycle}_{\text{Diff}}$ occasionally produces false positives, such as overly small distances for DeciDiffusion and other models, falling below the threshold. While DDPM_{inv} is relatively more discriminative than $\text{Cycle}_{\text{Diff}}$, it still lags behind ours.

Uniqueness analysis in the output domain Verification in fingerprint domain requires applying **FingerInv** to the suspicious model, which requires access to denoiser gradients and thus additional white-box privileges. In contrast, direct verification of the output image avoids this requirement and is more convenient. Previous statistical thresholding methods [25] assume that the error elements of two samples follow the i.i.d. Gaussian distribution with a manually estimated variance, potentially compromising threshold reliability. Using QR code images for direct scanning can simplify it, so we generate QR images with various fingerprint triggers and DDPMs, and create confusion matrices. In the confusion matrices, only the diagonal elements represent successful matches, leading to scannable QR codes. As shown in Figures 8, matched triggers and DDPMs produce clear, scannable QR code images with various l_{qr}, while mismatched pairs result in unscannable images, highlighting our method's strong discriminability.

Moreover, we compare the output discriminability for different inversion methods in Table 1, which presents the cross-verification results between different DDPMs using their triggers. The successful scanning is indicated by ✓, while ✗ means the failure. Ideally, the matrix should display ✓ only along the diagonal, with all non-diagonal elements marked as ✗, indicating that detections align correctly with their corresponding fingerprints and DDPMs. It is evident that $\text{Cycle}_{\text{Diff}}$ and DDPM_{inv} exhibit a significant risk of false positives in various scenarios, while our method demonstrates excellent discriminability, successfully distinguishing between all situations.

More analysis for uniqueness Recent work [12] showed that blind Gaussian DNNs can generate high-quality images using score-based reverse diffusion algorithms; and with sufficient training samples, two non-overlapping subsets can yield DNNs with nearly identical score functions and densities. Although these models are score-based generative models (SGMs), they are also de-

Table 1. Discriminative comparative results for output verification. Our fingerprint approach perfectly distinguishes different models (only the diagonal is ✓), while other baseline methods show varying degrees of misclassifications.

	PS-DDPMs					LDMs		
	Bedroom	Cat	CelebA	Church		SD	Pixart	Deci
CycleDiff								
Bedroom	✓	✓	✗	✗	SD	✓	✗	✗
Cat	✗	✓	✗	✓	Pixart	✓	✓	✓
CelebA	✓	✓	✓	✗	Deci	✓	✓	✓
Church	✓	✓	✗	✓	-	-	-	-
DDPM_{inv}								
Bedroom	✓	✓	✗	✓	SD	✓	✗	✗
Cat	✓	✓	✗	✓	Pixart	✓	✓	✗
CelebA	✓	✓	✓	✓	Deci	✗	✗	✓
Church	✓	✓	✗	✓	-	-	-	-
Ours								
Bedroom	✓	✗	✗	✗	SD	✓	✗	✗
Cat	✗	✓	✗	✗	Pixart	✗	✓	✗
CelebA	✗	✗	✓	✗	Deci	✗	✗	✓
Church	✗	✗	✗	✓	-	-	-	-

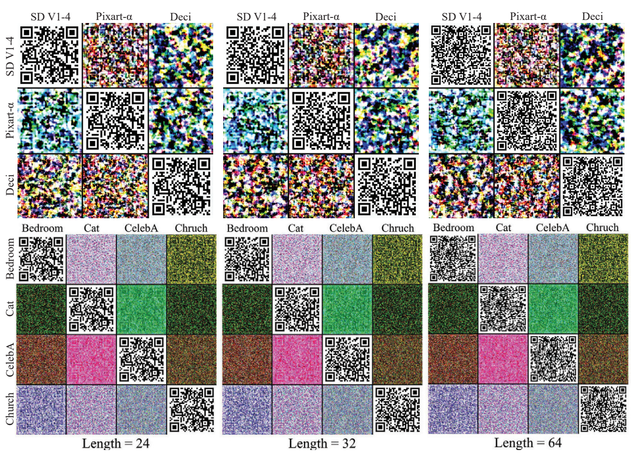

Figure 8. Results of discriminability analysis on output images.

noising ones, suitable for our method. Similar to Algorithm 1, we adapted our **FingerInv** for these SGMs (detailed in Section 9.2 of supplementary material).

We employed their pretrained denoisers, trained with 10K and 100K samples and resulting in similar score functions, to validate our **FingerInv**. These training samples are 80×80 grayscale facial images, and the QR codes images we used were at the same resolution with $l_{qr} = 16$. The highly similar models were open-sourced[3]. As shown in Figure 7, our method displayed discriminative capability and successfully reconstructed verification images. This indicates our method's strong fingerprint uniqueness and potential for extending to other DM variations.

4.3. Robustness Analysis

Attack settings We unified the attack settings for comparison on robustness. For PS-DDPMs, we applied an 8% pruning rate, conducted 1K fine-tuning iterations (by LAION-Art), and used float16 quantization. LDMs, particularly SD, benefit from a robust ecosystem for fine-tuning. We utilized pretrained models from the open-source community, including SD V1-5, Deliberate[4], Realistic Vision V2[5], and Anything V4[6]. These models, fine-tuned for specific purposes, enhance our analysis. We conducted a 50% pruning attack on SD and 10% on Pixart and DeciDiffusion. For quantization, float16 was used for SD and Pixart-α, while bfloat16 was used for DeciDiffusion.

For more visual results, we applied a wider range and stronger attacks, such as a 10% pruning rate for PS-DDPMs, a 15% pruning rate for DeciDiffusion and Pixart-α, and bfloat16 quantization for other DDPMs.

Impact of attacks To evaluate the impact of attacks on model performance, we generated 100 samples from both source and attacked models using a fixed random seed, and assessed them with PSNR, SSIM [36], LPIPS [42], and FID [7]. Figure 9 shows that even 5% pruning significantly reduces PSNR (some cases below 20 dB), SSIM (some cases below 0.8), and increases LPIPS (some cases above 0.6). Fine-tuning with different data distributions greatly affects FID, sometimes exceeding 1200. Quantization with bfloat16 also reduces PSNR (some cases below 20 dB) and SSIM (some cases below 0.8). These attack intensities constitute significant perturbations.

Figure 9. Performance variations across different attack scenarios.

Results of robustness analysis Figure 10 presents our visual results and shows clear QR code images gener-

[3] https://github.com/LabForComputationalVision/memorization_generalization_in_diffusion_models

[4] https://huggingface.co/XpucT/Deliberate

[5] https://huggingface.co/SG161222/Realistic_Vision_V2.0

[6] https://huggingface.co/xyn-ai/anything-v4.0

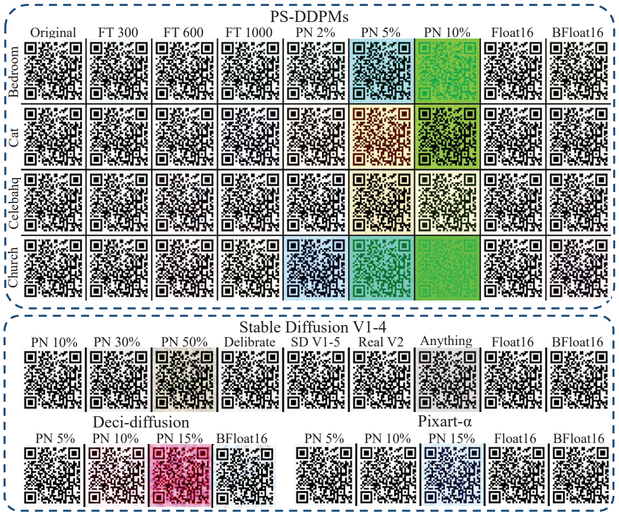

Figure 10. Visual results of robustness analysis.

ated under varying attacks. Table 2 compares our robustness with baselines. Our approach effectively detected the original QR images under various attacks for both PS-DDPMs and LDMs, outperforming non-invasive methods and matching the robustness of invasive techniques, which is comparable to watermarking methods. In addition, our method is resilient to attacks as described in [9] due to its non-invasive approach to the decoder of LDMs, surpassing $\text{Stable}_{\text{sig}}$ in model IP protection. Besides, our non-invasive method preserves the original model performance without additional fine-tuning or retraining, which offers significant advantages over invasive watermarking methods and supports a wider range of applications.

5. Conclusion and Discussion

We propose the first non-invasive fingerprinting method for DDPMs by modifying the noise to create distinctive fingerprint latent space, enabling fingerprint-verification pairs. Our method differentiates DDPMs with black-box access to denoisers, without altering model parameters or output quality. Experiments show strong distinctiveness and robustness for PS-DDPMs and LDMs, positioning our method as a promising solution for DDPM IP protection.

However, when considering the DDPM process as a whole, our method does not constitute a strictly black-box approach. The validation process requires manual input of latent components at each timestep during DDPM sampling. This can be inconvenient for direct validation in some fully encapsulated DDPM environments, such as, serving as an application programming interface (API). However, compared to previous fingerprint protection methods for image restoration, our approach significantly reduces permission requirements, as it does not necessitate white-box access to the denoisers during the verification stage.

Our future work will focus on fingerprinting based only on x_T for API applications and explore more properties of crossing route, including their uniqueness and extensions to other variants of diffusion models.

Table 2. Robustness results for various IP protection methods. We present verification results for various attacks and their success rates. For each method, we also include features such as non-invasiveness and theoretical robustness against [9].

	Eval	$\text{Cycle}_{\text{Diff}}$	DDPM_{inv}	$\text{Stable}_{\text{sig}}$	WM_{DM}	Ours
Pruning	Bedroom	✗	✗	-	✓	✓
	Cat	✓	✓	-	✓	✓
	CelebA	✓	✓	-	✓	✓
	Church	✓	✗	-	✓	✓
	SD V1-4	✓	✗	✓	✓	✓
	Deci	✗	✗	✓	✓	✓
	Pixart	✓	✓	✓	✓	✓
Finetuning	Bedroom	✓	✓	-	✓	✓
	Cat	✓	✓	-	✓	✓
	CelebA	✓	✓	-	✓	✓
	Church	✓	✓	-	✓	✓
	SD V1-5	✓	✓	✓	-	✓
	Delibrate	✓	✓	✓	-	✓
	Realistic	✓	✓	✓	-	✓
	Anything	✗	✓	✓	-	✓
Quantization	Bedroom	✓	✓	-	✓	✓
	Cat	✓	✓	-	✓	✓
	CelebA	✓	✓	-	✓	✓
	Church	✓	✓	-	✓	✓
	SD V1-4	✓	✓	✓	✓	✓
	Deci	✗	✗	✓	✓	✓
	Pixart	✓	✓	✓	✓	✓
Success Rate		81.82%	77.27%	100.00%	100.00%	100.00%
Non-invasive?		✓	✓	✗	✗	✓
Robust to [9]?		✓	✓	✗	✓	✓

Acknowledgements

This work is partially supported by Alibaba Cloud through the Research Talent Program with South China University of Technology; National Key Research and Development Program of China (No. 2024YFE0105400); National Natural Science Foundation of China (Nos. 62372186 and 62472179); Natural Science Foundation of Guangdong Province, China (No. 2023A1515012841); Fundamental Research Funds for the Central Universities, China (Nos. x2jsD2230220 and x2js/D2240750); and Singapore MOE AcRF Tier 1 (Grant No. A-8000981-00-00).

We would like to express our deepest respect and gratitude to our co-author Quan Yuhui, who sadly passed away in January 2025. His contributions to this work were substantial, and we are honored to include him as a co-author. This paper is dedicated to his memory.

References

[1] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX security symposium (USENIX Security 18)*, pages 1615–1631, 2018. 1

[2] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Ipguard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary. In *Proceedings of the 2021 ACM asia conference on computer and communications security*, pages 14–25, 2021. 1, 3

[3] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023. 1, 5

[4] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22466–22477, 2023. 1, 2, 3, 5

[5] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015. 3

[6] Zecheng He, Tianwei Zhang, and Ruby Lee. Sensitive-sample fingerprinting of deep neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4729–4737, 2019. 1, 3

[7] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 7

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2, 5

[9] Yuepeng Hu, Zhengyuan Jiang, Moyang Guo, and Neil Gong. Stable signature is unstable: Removing image watermark from diffusion models. *arXiv preprint arXiv:2405.07145*, 2024. 2, 3, 8

[10] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly DDPM noise space: Inversion and manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2, 3, 4, 5

[11] Wenbo Jiang, Hongwei Li, Guowen Xu, Tianwei Zhang, and Rongxing Lu. A comprehensive defense framework against model extraction attacks. *IEEE Transactions on Dependable and Secure Computing*, 21(2):685–700, 2023. 1

[12] Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and St'ephane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representation. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 6, 3, 4

[13] Tero Karras. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 5

[14] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024. 1, 4

[15] Erwan Le Merrer, Patrick Perez, and Gilles Trédan. Adversarial frontier stitching for remote neural network watermarking. *Neural Computing and Applications*, 32(13):9233–9244, 2020. 1

[16] Yugeng Liu, Zheng Li, Michael Backes, Yun Shen, and Yang Zhang. Watermarking diffusion model. *arXiv preprint arXiv:2305.12502*, 2023. 1, 2, 3

[17] Nils Lukas, Yuxuan Zhang, and Florian Kerschbaum. Deep neural network fingerprinting by conferrable adversarial examples. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 1, 3

[18] Takayuki Miura, Toshiki Shibahara, and Naoto Yanai. Megex: Data-free model extraction attack against gradient-based explainable ai. In *Proceedings of the 2nd ACM Workshop on Secure and Trustworthy Deep Learning Systems*, pages 56–66, 2024. 1

[19] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023. 3, 4

[20] Yuki Nagai, Yusuke Uchida, Shigeyuki Sakazawa, and Shin'ichi Satoh. Digital watermarking for deep neural networks. *International Journal of Multimedia Information Retrieval*, 7:3–16, 2018. 1

[21] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2

[22] Sen Peng, Yufei Chen, Cong Wang, and Xiaohua Jia. Intellectual property protection of diffusion models via the watermark diffusion process. *arXiv preprint arXiv:2306.03436*, 2023. 1, 2, 3

[23] Zirui Peng, Shaofeng Li, Guoxing Chen, Cheng Zhang, Haojin Zhu, and Minhui Xue. Fingerprinting deep neural networks globally via universal adversarial perturbations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13430–13439, 2022. 1, 3

[24] Yuhui Quan, Huan Teng, Yixin Chen, and Hui Ji. Watermarking deep neural networks in image processing. *IEEE transactions on neural networks and learning systems*, 32 (5):1852–1865, 2020. 1, 5

[25] Yuhui Quan, Huan Teng, Ruotao Xu, Jun Huang, and Hui Ji. Fingerprinting deep image restoration models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13285–13295, 2023. 1, 2, 3, 4, 5, 6

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 1, 2, 5

[27] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. Deepsigns: A generic watermarking framework for ip protection of deep learning models. *arXiv preprint arXiv:1804.00750*, 2018. 1

[28] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 5

[29] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 1

[30] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[31] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 1

[32] DeciAI Research Team. Decidiffusion 1.0, 2023. 1, 5

[33] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pages 601–618, 2016. 1

[34] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on international conference on multimedia retrieval*, pages 269–277, 2017. 1

[35] Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22532–22541, 2023. 3, 4

[36] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7

[37] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[38] Chen Henry Wu and Fernando De la Torre. A latent space of stochastic diffusion models for zero-shot image editing and guidance. In *ICCV*, 2023. 2, 3, 4, 5

[39] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 5

[40] Zihan Yuan, Li Li, Zichi Wang, and Xinpeng Zhang. Watermarking for stable diffusion models. *IEEE Internet of Things Journal*, 2024. 1, 2, 3

[41] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia conference on computer and communications security*, pages 159–172, 2018. 1

[42] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 7

[43] Jingjing Zhao, Qingyue Hu, Gaoyang Liu, Xiaoqiang Ma, Fei Chen, and Mohammad Mehedi Hassan. Afa: Adversarial fingerprinting authentication for deep neural networks. *Computer Communications*, 150:488–497, 2020. 1, 3

[44] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai-Man Cheung, and Min Lin. A recipe for watermarking diffusion models. *arXiv preprint arXiv:2303.10137*, 2023. 1, 2, 3, 5