Safe-SD: Safe and Traceable Stable Diffusion with Text Prompt Trigger for Invisible Generative Watermarking

Anonymous Authors
Submission Id: 3862

ABSTRACT

Recently, stable diffusion (SD) models have typically flourished in the field of image synthesis and personalized editing, with a range of photorealistic and unprecedented images being successfully generated. As a result, widespread interests have been ignited to develop and use various SD-based tools for visual content creations. However, the exposures of AI-created contents on public platforms could raise both legal and ethical risks. In this regard, the traditional methods of adding watermarks to the already generated images (i.e. *post-processing*) may face a dilemma (e.g., being *erased* or *modified*) in terms of copyright protection and content monitoring, since the powerful image inversion and text-to-image editing techniques have been widely explored in SD-based methods. In this work, we propose a **Safe** and high-traceable **S**table **D**iffusion framework (namely **Safe-SD**) to adaptively implant the graphical watermarks (e.g., *QR code*) into the imperceptible structure-related pixels during generative diffusion process for supporting text-driven invisible watermarking and detection. Different previous high-cost *injection-then-detection* training framework, we design a simple and unified architecture, which makes it possible to simultaneously train watermark injection and detection in a single network, greatly improving the efficiency and convenience of use. Moreover, to further support text-driven generative watermarking and deeply explore its robustness and high-traceability, we elaborately design a λ-sampling and λ-encryption algorithm to fine-tune a latent diffuser wrapped by a VAE for balancing high-fidelity image synthesis and high-traceable watermark detection. We present our quantitative and qualitative results on two representative datasets LSUN, COCO and FFHQ, demonstrating state-of-the-art performance of Safe-SD and showing it significantly outperforms the previous approaches.

CCS CONCEPTS

• **Security and privacy** → **Digital rights management**; • **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

Invisible Watermarking, Generative Copyright, Stable Diffusion

ACM Reference Format:
Anonymous Authors and Submission Id: 3862. 2024. Safe-SD: Safe and Traceable Stable Diffusion with Text Prompt Trigger for Invisible Generative Watermarking. In *Proceedings of the 32nd ACM International Conference on Multimedia (MM'24), October 28-November 1, 2024, Melbourne, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

"In art, what we want is the certainty that one spark of original genius shall not be extinguished."

– Mary Cassatt

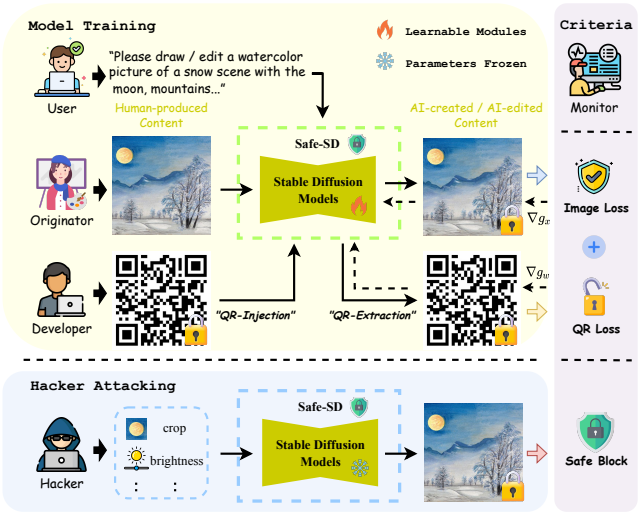


Figure 1: The overview of our proposed Safe-SD framework. In which, different humans indicate the different roles being simulated in the AIGC environment such as *user, originator, developer, hacker* and *monitor*.

Recent years has witnessed the remarkable success of diffusion models [21, 44, 51], due to its impressive generative capabilities. After surpassing GAN on image synthesis [11], diffusion models have shown a promising algorithm with dense theoretical founding, and emerged as the new state-of-the-art among the deep generative models [18, 22, 29, 38, 43, 47, 50, 52, 53, 57, 64]. Notably, Stable Diffusion [48], as one of the most popular and sought-after generative models, has sparked the interests of many researchers, and a series of SD-based works have been proposed and exploited to produce plenty of AI-created or AI-edited images, such as ControlNet [67], SDEdit [40], DreamBooth [49], Imagic [27], InstructPix2Pix [3] and Null-text Inversion [41], which raises profound concerns about ethical and legal risks for AI-generated content (AIGC) being unscrupulously exposed on public platforms and raises new challenges for copyright protection and content monitoring.

These concerns may be elaborated into the following three aspects: (1) ***Originator Concern.*** An artistic work or photograph produced by the original author may be edited or modified at will by AI today and published to the public platform for commercial profit, which infringes on the interests of the originator. Take Figure 1 as an example, when a wonderful hand-crafted watercolor painting is published online by the originator, another user could download it without any restrictions and then request the SD-based model to edit the artwork through an accompanying prompt *"please edit a watercolor picture of..."*, whereas ultimately attributes the AI-created production and its ancillary value to the user and the given prompt, which may have violated the rights of the originator. If this is an

commercial advertisement or model shooting, product designs or industrial drawings, etc., it may cause more serious infringement of interests. (2) ***Developer Concern.*** Which means the potential risks that SD-based tools open sourced by developers may be abused by people with bad motives to engage in underground activities, such as fake news fabrication, political rumors publishing or pornographic propaganda, etc., simply by editing human characteristics (*e.g.,* replacing faces). (3) ***Monitor Concern.*** Which means it's extremely difficult for the monitor of online platforms to distinguish which visual contents are produced by AI and judge whether it should be safely blocked to ensure their compliance with legal and ethical standards, since the fidelity and texture of AI-created images have approached human levels. For example, a generated picture recently have won an art competition [17], which suggests humans will soon be unable to discern the subtle differences between AI-generated content and human-created content. Overall, the above concerns illustrate the fact that the emergence of powerful AI-generative tools and the lack of traceability of their generated productions may open the door to new threats such as artwork plagiarism, copyright infringement, political rumors publishing, and portrait rights infringement and so on.

To cope with the above concerns, we propose a ***Safe*** *and high-traceable* ***S****table* ***D****iffusion* framework with a text prompt trigger for unified generative watermarking and detection, *Safe-SD* for short. Note that since Stable Diffusion [48] is an open source model with most ecologically complete as well as widely used foundation models and has been applied to numerous generative tasks, we only focus on the SD-based models for invisible watermark injection and extraction, which can be further easily extended to other diffusion models such as DALL-E2 [47], Imagen [50] and Parti [64] by only replacing the weights and bias of the U-Net's parameters in diffusion models and adding a lightweight inject-convolution layer from our Safe-SD. Different from existing methods that *post-processing* [8], *injection-then-detection* [65] or are based solely on *decoder fine-tuning* [15], our proposed models have the following new features:

- Designing a unified watermarking and tracing framework, which makes it possible to simultaneously train watermark injection and detection in a single network to balance high-fidelity image synthesis and high-traceable watermark detection, greatly improving the training efficiency and convenience of use.
- Enabling to implant the graphical watermarks (e.g., *QR code*) into the imperceptible structure-related pixels, which ties the pixels of watermark to each diffusion step for high-robustness, unlike *post-processing* methods, may be easily erased or modified by image inversion or editing models.
- Supporting text-driven image watermarking and multi-watermarking scenarios, which can be applied to a wider range of downstream tasks such as: *text-to-image* synthesis, *text-based* image editing, *multi-watermarks* injection, etc.

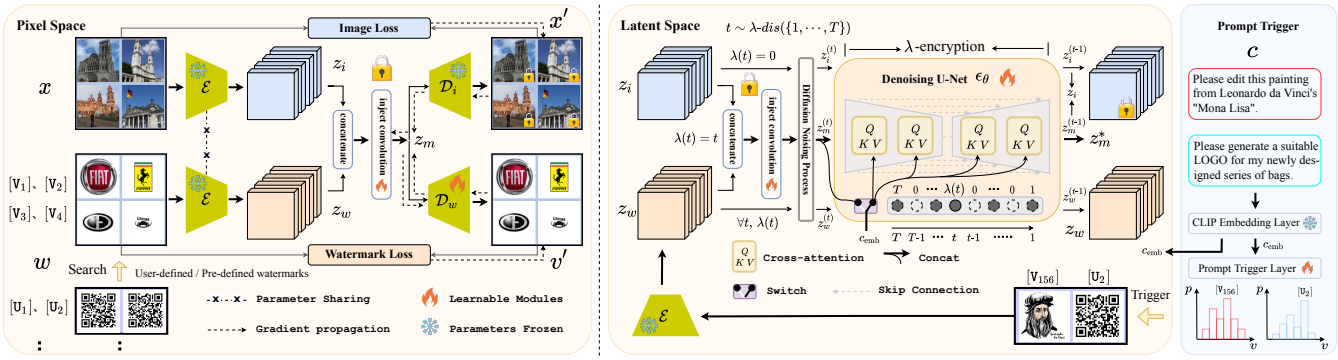
Experiments on three representative datasets LSUN-Churches [63], COCO [36], FFHQ [25] demonstrate the effectiveness of Safe-SD, showing that it achieves the *state-of-the-art* generative results against previous invisible watermarking methods. Further qualitative evaluations exhibit the pixel-wise differences between the original images and watermarked images, and the robustness study quantitatively evaluates the anti-attack ability, which further verifies the superiority of Safe-SD in balancing high-resolution image synthesis and high-traceable watermark detection.

2 RELATED WORK

Diffusion Models. Recent years has witnessed the remarkable success of diffusion-based generative models, due to their excellent performance in the diversity and impressive generative capabilities. These previous efforts mainly focus in sampling procedure [37, 53], conditional guidance [11, 43], likelihood maximization [28, 29] and generalization ability [18, 26] and have enabled state-of-the-art image synthesis. Stable Diffusion [48] is one of the most widely used diffusion models, due to its open source and user-friendly features, it has recently gained great attention and become one of the leading researches in image generation and manipulation.

Image Watermarking Techniques. To trace copyright and make AI-generated content detectable, numerous watermarking techniques have been proposed for deep neutral networks [1, 31, 32, 34, 35, 39, 42], which can basically be classified into two categories: discriminative models and generative models. In discriminative models, watermarking techniques are mainly dominated by white-box or black-box models. The white-box models [4, 7, 13, 33, 55, 56, 59, 61] need access to the models and their parameters (white-box access) in order to extract the watermarks, while the black-box models [5, 10, 20, 23, 54, 62, 66, 68] only adopt predefined inputs as triggers to query the models (black-box access) without caring about their internal details. In generative models, the previous methods mainly investigate GANs by watermarking all generative images [9, 14, 45, 70] such as binary strings embedding [14, 65, 70], textual message encoding [9] and graphic watermark injection [45]. Very recently, some researchers [15, 24, 69] have extended binary strings embedding technique into diffusion-based architecture for digital copyright protection, one of the most representative digital watermark injection methods is Stable Signature [15]. However, binary digital watermarking suffers from erasuring and overwriting threats when meeting with DDIM inversion [51], overwriting attacks [60] and backdoor attacks [6, 19].

Different from them, we explore a more secure and efficient diffusion-based generative framework *Safe-SD*, with imperceptible watermark injection module and textual prompt trigger, which is designed in a unified watermarking and tracing framework, making it possible to simultaneously achieve watermark injection and detection in a single network, greatly improving the training efficiency and convenience of use for multimedia and AIGC community. For security, the Safe-SD enables SD-based generative network to implant the graphical watermarks (e.g., QR code) into the imperceptible structure-related pixels and retain high-fidelity image synthesis and high-traceable watermark detection capabilities, which is hard to be erased or modified as the graphical watermark is tightly bound to the progressive diffusion process. For robustness, we introduce a fine-tuned latent diffuser with an elaborately designed λ-encryption algorithm for high-traceable watermarking training. Moreover, we also conduct a hacker attacking study (Sec. 4.5), by setting up 5 attack tests to evaluate the robustness of proposed Safe-SD against attacks. Note our Safe-SD methods can be easily extended to other diffusion-based models such as DALL-E2 [47],



(a) **Stage1:** Pre-train graphic watermark injector/extractor

(b) **Stage2:** Fine-tune latent λ-encryption diffuser with prompt triggers

Figure 2: The framework of Safe-SD model.

Imagen [50] and Parti [64] by only replacing the weights and bias of the U-Net's parameters in diffusion models and adding a light-weight *inject-convolution* layer have pretrained in our Safe-SD.

3 METHOD

As depicted in Figure 2, Safe-SD mainly contains two stages: 1) Pre-training stage for unified watermark injector/extractor (Sec.3.1) and 2) Fine-tuning stage for latent diffuser with text prompt trigger (Sec.3.2). The former aims to train a modified SD's *first-stage-model* (with a brand new dual variational autoencoder) to obtain a unified graphic watermark injection and extraction network, whereas the latter serves as a latent diffuser with an elaborately designed temporal λ-encryption algorithm for more secure and high-traceable watermark injection. Moreover, we introduce a novel prompt triggering mechanism to support text-driven image watermarking and copyright detection scenes.

During inference, the pipeline of our proposed model is: 1) Safe-SD first accepts a text condition c and an image x ({"`image synthesis`": $x = \varnothing$; "`image editing`": x}) as inputs, and then the prompt trigger $p(\cdot)$ determines which watermark w should be injected based on the given condition c. Meanwhile Safe-SD randomly allocates a key $m \in \{0,1\}^T$ (T is diffusion steps) into the next step; 2) The encoder $\mathcal{E}$ of the *first-stage-model* first encodes the image x and watermark w into latent variables z_i and z_w respectively and then feeds them immediately into the second stage; 3) The latent diffuser first accepts the latent variables z_i and z_w, condition c and the key m, then performs temporal λ-encryption algorithm (**Algorithm** 1) for high-traceable watermark injection or performs condition-guided invert denoising (**Algorithm** 2) for high-fidelity image synthesis; 4) The decoder $\mathcal{D}_i$ of the *first-stage-model* then serves as a watermarker to generate the above watermarked images with λ-encryption for safe readout, and another decoder $\mathcal{D}_w$ serves as a detector to decode the injected watermark hidden from the images for detection, authentication and copyright trace.

3.1 Pre-training watermark injector/extractor

Our *first-stage-model* is designed to jointly train a watermark extractor $\mathcal{D}_w$ and an image generator $\mathcal{D}_i$ with invisible watermarking when they are equally fed the latent variables z_m of an image mixed with watermark features. Since it is fully pre-trained to balance the two goals of simultaneously generating high-quality images and clear watermarks, this *first-stage-model* can adapt to accept any latent mixture z_m^* with λ-encryption watermarking in the second stage, to ultimately complete the dual decodings. Details of the *first-stage-model* are introduced below.

Shared graphic encoder. Given an input image x and a randomly searched watermark w, $x, w \in \mathbb{R}^{H\times W\times 3}$. The shared graphic encoder $\mathcal{E}$ first projects the image x and watermark w into latent variables z_i and z_w, *i.e.*,, $z_i = \mathcal{E}(x), z_w = \mathcal{E}(w), z_i, z_w \in \mathbb{R}^{h\times w\times d}$, where h and w respectively denote scaled height and width (default scaled factor $f = H/h = W/w = 8$), and d is the dimensionality of the projected latent variables.

Injection convolution layer. Safe-SD first concatenates the projected image z_i and watermark z_w in the channel dimension, and then obtains the mixture features $z_m \in \mathbb{R}^{h\times w\times d}$ through a simple injection convolution layer $f_c(\cdot) : \mathbb{R}^{h\times w\times 2d} \to \mathbb{R}^{h\times w\times d}$. Formally,

$$z_m = f_c(z_i, z_w) \tag{1}$$

Dual goal decoders. To synchronously train a image generator $\mathcal{D}_i$ with invisible watermarking and a watermark extractor $\mathcal{D}_w$, we introduce a dual decoding mechanism with two decoder-copies from SD's *first-stage-model* (i.e., *vae* [12]), and one copy with frozen parameters θ_f and the other copy with trainable parameters θ_t. Note that since decoder $\mathcal{D}_i$ plays the role of an image generator with an invisible watermark injection and has been fed to the mixture variable z_m, it needs to be assigned to the frozen parameter θ_f for watermarking image generation, while decoder $\mathcal{D}_w$ only serves as a watermark extractor (also with the mixed variables z_m as input), therefore need to be assigned trainable parameters θ_t for watermark extraction. Formally,

$$\hat{x} = \mathcal{D}_i(z_m; \theta_f), \ \hat{w} = \mathcal{D}_w(z_m; \theta_t) \tag{2}$$

To maximize the accuracy of watermark extraction and enabling to generate high-resolution images, we set up a weighting-based loss $\mathcal{L}_{s^1}$ to supervise the entire *first-stage-model*, which can be formally represented as,

$$\mathcal{L}_{s^1} = ||x - \hat{x}||^2 + \gamma \cdot ||w - \hat{w}||^2 + \mathcal{L}_{adv} \tag{3}$$

where γ is the weighting hyperparameter (default γ equals 1), and $\mathcal{L}_{adv}$ denotes the adversarial training loss, which maintains the same setting as in VQGAN [12].

Algorithm 1 λ-sampling based forward diffusion

Input: Latent image z_i and watermark z_w, diffusion steps T
Output: λ-watermarking noise z_m^T, key m

$t \sim \lambda\text{-}dis(t) = \{\underbrace{1, 3, ..., T}_{\lambda}, \underbrace{0, ..., 0}_{T\text{-}\lambda}\}$;
$t \rightarrow \lambda(t)$;
for t = 1,2,...,T **do**
  if $\lambda(t) = 0$ **then**
    $z_i^{(t)} = \alpha_t \cdot z_i^{(t-1)} + \sqrt{1-\alpha_t^2}\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$;
    $z_i^{(t)} \rightarrow z_m^{(t)}$
    $m_t = 0$;
  else if $\lambda(t) = t$ **then**
    $f_c(z_i^{(t-1)}, z_w^{(t-1)}) \rightarrow z_m^{(t-1)}$;
    $z_m^{(t)} = \alpha_t \cdot z_m^{(t-1)} + \sqrt{1-\alpha_t^2}\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$;
    $m_t = 1$;
  end if
  $z_w^{(t)} = \alpha_t \cdot z_w^{(t-1)} + \sqrt{1-\alpha_t^2}\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$;
end for
$\text{Iter}^+(z_m^{(t)}) \rightarrow z_m^T$;
$\text{Compose}(m_t) \rightarrow m$;
return $\{z_m^T, m\}$

3.2 Fine-tuning latent λ-encryption diffuser

The *second-stage-model* mainly serves as a temporal *λ-encryption* diffuser with prompt triggering mechanism, which mainly relies on a temporal injection algorithm by accepting a binary key $m \in \{0,1\}^T$ as *instruction-code* to control whether each diffusion step requires performing watermark injection, for cryptographic image synthesis with minor structural changes. Details of the *second-stage-model* are as follows.

Prompt trigger. The prompt trigger is designed to achieve non-sensitive watermark triggering, which accepts a textual *editing*- or *synthesis*-related instruction as input, by following a CLIP embedding layer and a linear prompt trigger layer, to ultimately obtain a watermark (*predefined* or *user-defined* watermark) with the highest probability for subsequent invisible watermark injection. Moreover, for stable copyright protection, Safe-SD can also support watermark injection based on special instructions, such as when given the instruction: "*Please help me edit this personal photo with my avatar watermark* [U]" and the accompanying avatar "[U]" as a personalized watermark, Safe-SD can be triggered directly with this specified watermarking LOGO. Note that in our experiments, we adopt a public LOGO dataset [1] to represent pre-defined or user-defined watermarks for the training of the Safe-SD.

Forward diffusion with λ-sampling. To enable the watermark to be adaptively injected into the image synthesis process with temporal diffusion and to maintain traceability, we propose the forward diffusion with λ-sampling. We first introduce the definitions of *λ-sampling* and *λ-distribution* below, and then explain how it can be used for watermark injection based on temporal encryption.

First, for a given sequence $(x_1, ..., x_N)$, the *λ-sampling* operation is defined as: randomly selecting λ elements from the sequence with N elements for sampling, and at the same time, the unsampled elements are set to 0. Thereafter the obtained discrete distribution is

[1] https://github.com/msn199959/Logo-2k-plus-Dataset

Algorithm 2 λ-encryption based inversion denoising

Input: Latent image z_i and watermark z_w, denoising key m
Output: λ-encrypted mixture z_m^0, latent image z_i^0 and watermark z_w^0

for t = T, $T-1$,...,1 **do**
  if $m_t = 0$ **then**
    $z_i^{(t-1)} = \sqrt{\alpha_{t-1}}\left(\frac{z_i^{(t)} - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(z_i^{(t)}, c, t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta^{(t)}(z_i^{(t)}) + \sigma_t\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$;
  else if $m_t = 1$ **then**
    $z_m^{(t-1)} = \sqrt{\alpha_{t-1}}\left(\frac{z_m^{(t)} - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(z_m^{(t)}, c, t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta^{(t)}(z_m^{(t)}) + \sigma_t\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$;
  end if
  $z_w^{(t-1)} = \sqrt{\alpha_{t-1}}\left(\frac{z_w^{(t)} - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(z_w^{(t)}, c, t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \epsilon_\theta^{(t)}(z_w^{(t)}) + \sigma_t\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$;
end for
$\text{Iter}^-(z_i^{(t)}, z_w^{(t)}) \rightarrow (z_i^0, z_w^{0,i})$;
$\text{Iter}^-(z_m^{(t)}, z_w^{(t)}) \rightarrow (z_m^0, z_w^{0,m})$;
$z_w^{0,m} \rightarrow z_w^0$ **if** $m_0 = 1$ **else** $z_w^{0,i} \rightarrow z_w^0$;
return $\{z_m^0, z_i^0, z_w^0\}$.

referred to as the "*λ-distribution*" corresponding to this *λ-sampling*, abbreviated as $\lambda\text{-}dis(\cdot)$, where,

$$\lambda\text{-}dis(i) = \begin{cases} x_i & \text{if } x_i \text{ is sampled,} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Then, we introduce this *λ-sampling* based temporal encryption mechanism, which aims to bind a given watermark w to a diffusion synthesis process $q(z_m^{(t)}|z_i^{(t-1)}, z_w^{(t-1)})$ and simultaneously generate a binary key m for traceability, as illustrated in Algorithm 1. As shown in Figure 3, when $\lambda(t)$ equals t, the Safe-SD is activated to perform the watermark injection process through a temporal injection cell (right side of Figure 3), which is consistent with the *first-stage-model* to ensure good generalization for watermark injection and can be formally described as,

$$z_m^{(t-1)} = f_c(z_i^{(t-1)}, z_w^{(t-1)}) \tag{5}$$

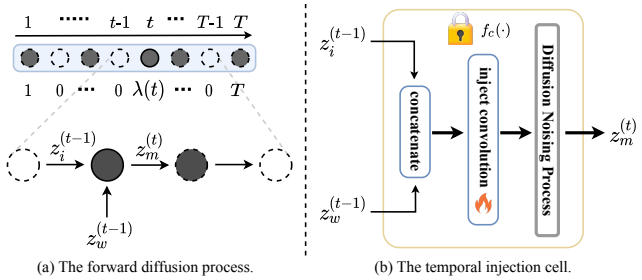
$$z_m^{(t)} = \alpha_t \cdot z_m^{(t-1)} + \sqrt{1-\alpha_t^2}\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{6}$$

where $f_c(\cdot)$ denotes an aforementioned learnable injection convolution layer proposed by us for mapping the concatenation of the latent image $z_i^{(t-1)}$ and watermark $z_w^{(t-1)}$ into a latent watermarking mixture $z_m^{(t-1)}$. Whereas, when $\lambda(t)$ equals 0, the Safe-SD performs this forward diffusion simply by adding random noise $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$ to the latent vector $z_i^{(t-1)}$ of the image from the previous step, formally,

$$z_m^{(t)} = \alpha_t \cdot z_i^{(t-1)} + \sqrt{1-\alpha_t^2}\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{7}$$

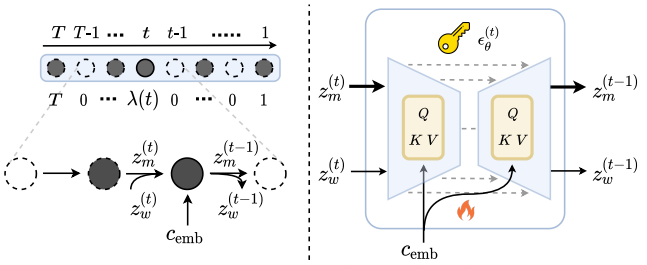
Note Safe-SD uses a binary value of 0 or 1 to record this forward diffusion process with *λ-sampling* and then to compose them into a binary key $m \in \{0,1\}^T$, which will serve as readout to control the subsequent inverted denoising.

Inverted denoising based λ-encryption. To fine-tune the latent diffuser from *second-stage-model* to enable the input image,



(a) The forward diffusion process. (b) The temporal injection cell.

Figure 3: The forward diffusion with λ-*sampling* watermarking.



(a) The inverted denoising process. (b) The conditional denoising diffuser.

Figure 4: The inverted denoising based λ-*encryption* prediction.

watermark and their latent mixture to be correctly denoised by an U-Net network, and to ultimately ensure high-fidelity image synthesis and watermark extraction, we propose this inverted denoising module based λ-*encryption*. Consistent with the forward process mentioned above, this inverted denoising module is controlled by an *if-else*-branched Markov chain, which is recorded by the binary key m (*e.g.,* 10101101) generated above. Similarly, we first introduce the λ-*encryption* mechanism below, and then explain how it can be used for inverted denoising.

First, for a given sequence $(x_1, ..., x_N)$ and a key $m \in \{0,1\}^N$, this λ-*encryption* is defined as: at any position where $m_i = 1$, the original data x_i is modified into x_i^* by superimposing a perturbation x_Δ onto x_i (*i.e.,* $x_i^* = x_i \oplus x_\Delta$), while keeping the original data unchanged at other positions where $m_i = 0$, to finally obtain the encrypted sequence. The advantage of this λ-*encryption* method is that it maintains the distribution of the original data as much as possible while achieving controllable encryption.

Then, we introduce a λ-*encryption* based inverted denoising strategy, which treats the latent watermark z_w as a perturbation when $m = 1$, and z_w is subsequently superimposed on the latent variable z_i of the image (*i.e.,* $z_m^* = z_i \oplus z_w$) by an injection convolution layer $f_c(\cdot)$ to ultimately obtain a watermarked image (*i.e.,* encrypted vector) in latent space, as shown in Figure 3(b). Formally,

$$z_m^{(t)} = z_i^{(t-1)} \oplus z_w^{(t-1)} = f_c(z_i^{(t-1)}, z_w^{(t-1)}) \tag{8}$$

Furthermore, as illustrated in Algorithm 2, when $m = 0$, the latent variable z_i of the original image is directly sent to U-Net for denoising without adding any disturbance. As shown in Figure 4, when denoising the perturbed image $z_m^{(t)}$, the watermark $z_w^{(t)}$ is simultaneously fed into U-Net for balancing image generation and watermark extraction. Note that this does not require using U-Net twice but simply by first concatenating them and then feeding them together into a shared U-Net network $\epsilon_\theta(\cdot)$ for denoising as,

$$(z_m^{(t-1)}, z_w^{(t-1)}) = \mathcal{S}_{ddim}\left(\epsilon_\theta^{(t)}(z_m^{(t)}, z_w^{(t)}|c, t)\right) \tag{9}$$

where $\mathcal{S}_{ddim}(\cdot)$ denotes the DDIM [51] sampling strategy executed during inference, which is sampled from the predicted $\epsilon_\theta^{(t)}$ to obtain the final $z_m^{(t-1)}$ and $z_w^{(t-1)}$ (through a tensor split operation `torch.chunk()`). Similarly, when an unperturbed image $z_i^{(t)}$ as input, the watermark $z_w^{(t)}$ is also sent to U-Net for denoising as,

$$(z_i^{(t-1)}, z_w^{(t-1)}) = \mathcal{S}_{ddim}\left(\epsilon_\theta^{(t)}(z_i^{(t)}, z_w^{(t)}|c, t)\right) \tag{10}$$

Fine-tuning objectives. To fine-tune this latent diffuser with λ-*sampling* and λ-*encryption* to adapt to the dual goal decoders from the *first-stage-model*, we set up a stepwise denoising loss,

$$\mathcal{L}_{s^2} = \underbrace{||\epsilon - \epsilon_\theta^{(t)}(z_m^{(t)}, z_w^{(t)})||_2^2}_{m_t=1} + \underbrace{||\epsilon - \epsilon_\theta^{(t)}(z_i^{(t)}, z_w^{(t)})||_2^2}_{m_t=0} \tag{11}$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ denotes standard Gaussian noise, which is consistent with Stable Diffusion [48]. Moreover, the classifier-free guidance technique [22] is also used in the training of Safe-SD.

4 EXPERIMENTS

4.1 Experimental Setting

Datasets. We follow [12] to pre-train the *first-stage-model* of our Safe-SD on LSUN-Churches [63], COCO [36], FFHQ [2] [25] and Logo-2K [58] datasets with image resolution 256×256, and further follow Dreambooth [49] to fine-tune the latent diffuser of the *second-stage-model* for $\lambda-$encrypted watermark injection. For the training of the text-conditional diffusion models, we follow [49] to leverage a textual prompt (*e.g., "a photo of a church with watermark* `[V]` *(or* `[U]`*)"*) as the guidance condition and adopt the graphical LOGOs from Logo-2K as pre-defined watermarks to finetune our Safe-SD model in our experiments. Specifically, 126, 227 images on training set of LSUN-Churches, 63, 000 images on training set of FFHQ and 167, 140 watermarks on Logo-2K are utilized to train the models. During testing, 1, 000 images and 1, 000 watermarks are randomly composed to perform the quantitatively experimental evaluations.

Implementation details. We follow SD [48] to resize all the images to a resolution of 256×256, and the batch size is set to 4. The scaling factor f is set to 8 and the guidance factor of the classifier-free is set to 7.5. During inference, the pre-trained CLIP embedding layer [46] is leveraged to match the suitable watermarks for adaptive prompt triggering strategy and DDIM [51] sampling is executed for final image synthesis. All the experiments are performed for 20 epochs on 2 NVIDIA RTX3090 GPUs with PyTorch framework and the optimization and schedule setups are consistent with [48].

4.2 Image generation quality for watermarking

Qualitative Evaluation. To evaluate the image generation quality and the fidelity with watermarking, we first conduct the qualitative experiments by visualizing the pixel-level differences (×10) between original image and watermarked image (marked as *W/. Watermark*), which are presented in Figure 5. Specifically, in Figure 5,

[2] https://github.com/NVlabs/ffhq-dataset

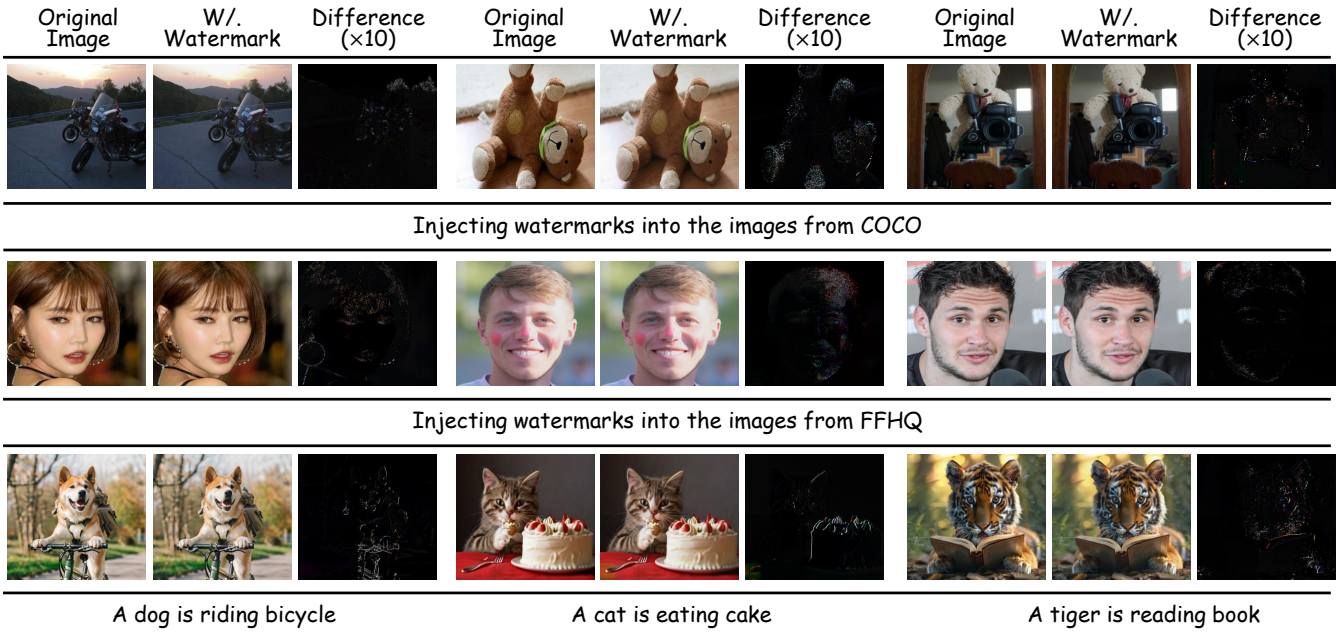


Figure 5: Evaluation the image quality by visualizing the pixel-level differences (×10) between original image and watermarked image (marked as W/. Watermark). Top: natural images from COCO [36]. Mid: facial images from FFHQ [25]. Bottom: text-generated images.

Methods	Type	PSNR ↑	FID ↓	LPIPS ↓	CLIP ↑
SSL Watermark [2022]	string	31.60	19.63	0.261	84.03
Baluja et.al. [2019]	graphics	30.41	20.39	0.317	83.63
FNNS [2021]	string	32.71	19.03	0.243	85.99
HiDDeN [2018]	string	32.99	19.49	0.244	85.43
Stable Signature [2023]	string	31.09	19.47	0.263	87.90
Safe-SD (Ours)	graphics	**33.17**	**18.89**	**0.232**	**88.15**

Table 1: The comparison results on LSUN-Churches dataset.

Methods	Type	PSNR ↑	FID ↓	LPIPS ↓	CLIP ↑
HiDDeN [2018]	string	32.19	19.58	0.217	93.32
Baluja et.al. [2019]	graphics	29.17	20.85	0.403	91.93
FNNS [2021]	string	31.96	19.56	0.220	92.00
SSL Watermark [2022]	string	30.47	19.91	0.262	91.51
Stable Signature [2023]	string	30.88	20.33	0.231	93.01
Safe-SD (Ours)	graphics	**32.73**	**19.36**	**0.215**	**93.99**

Table 2: The comparison results on FFHQ dataset [25].

we respectively test Safe-SD on natural images from COCO [36] (**Top**), facial images from FFHQ [25] (**Mid**), and text-generated images (**Bottom**). From Figure 5, we can observe that: **1) All watermarked images by our Safe-SD maintain high-fidelity.** Particularly, even for challenging facial images, the watermarked results still can finely preserve the details of hair. Moreover, combined with the results of Figure 6 (additionally presenting the detected watermarks), we can notice that our Safe-SD can simultaneously balance the quality of the detected watermarks and the watermarked images. It is worth noting that compared to previous digital watermarking methods [15, 69], our Safe-SD has higher fault tolerance. For example, when some several pixels are incorrectly predicted, it will not lead to incorrect detection and authentication in our method, but in the digital watermarking method, the incorrect prediction of every binary bit (*e.g.*, "0101"→"0111") may seriously affect the final identification result. **2) There are still subtle textured differences in enlarged pixel-level, but that's almost imperceptible and well ensures traceability.** According to the enlarged (×10) pixel-wise results, it can be observed that the generative differences mainly come from visual contents with dense texture, such as hair and eyes in facial images, but note that it is almost impossible to discern by the human eyes. That also reveals that the information hidden in the image cannot disappear, but can only be moved to an imperceptible location to ensure traceability. **3) Safe-SD is suitable for a wide variety of images and well supports text-driven generative watermarking.** As shown in Figure 5, the experiments are conducted on a wide variety of images, such as the natural images from COCO [36], facial images from FFHQ [25], and text-generated images (bottom), showing all the generated images watermarked by our Safe-SD maintain high-fidelity, which demonstrates the powerful generalization ability of our Safe-SD. Besides, Figure 6 presents more qualitative comparison results with previous graphical watermarking method [2], which further verifies the superiority of our model in balancing high-resolution image synthesis and high-traceable watermark detection.

Quantitative Evaluation. Following [15], we further quantitatively evaluate our approach in PSNR, FID, LPIPS and CLIP-Score metrics on LSUN-Churches and FFHQ datasets, which is shown in Table 1 and Table 2. From the results in the two tables, we can observe that our model Safe-SD achieves the *state-of-the-art* performance on all four metrics and obtains the best generative results, even with more challenging graphical watermarking, i.e.,

Figure 6: Qualitative comparison results. Note the first column is the "*original image*" and "*original watermark*" (upper right corner), the second column is the "*watermarked image*" using Baluja et.al. method [2] and the "*detected watermark*" (upper right corner). The third column is consistent with the second column but with our Safe-SD approach.

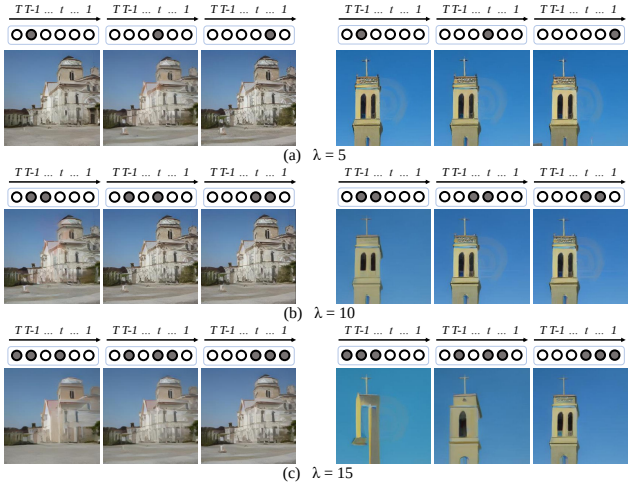


Figure 7: The effect of the λ. Two groups of instances are presented to explore the influence of the frequency and time period of λ-encryption. Note the solid ball denotes the current λ-$dis(t)$ is not 0.

directly interfering with pixels, compared to string-based methods [15, 16, 30, 70]. In particular, our model outperforms Stable Signature, a recent generative work, by 6.69%, 2.98%, 11.79% and 0.28% in four metrics on the LSUN-Churches dataset, and exceeds by 5.99%, 4.77%, 6.93% and 1.05% on the FFHQ dataset, which further verifies the superiority and effectiveness of Safe-SD.

4.3 Explore on λ-encryption watermarking

The frequency of λ-encryption. To deeply explore the performance of λ-encryption in image watermarking in our approach, we perform a study on the impact of watermarking frequency λ on image synthesis quality, as shown in Figure 7. It can be observed that with the increase of λ (*i.e.*, from 5 to 15), the performance of generated images may be affected due to the interference of watermark information, so we need to balance the frequency of watermark injection and the image's fidelity and finally choose $\lambda = 10$ (50 steps in total) as the appropriate watermarking frequency.

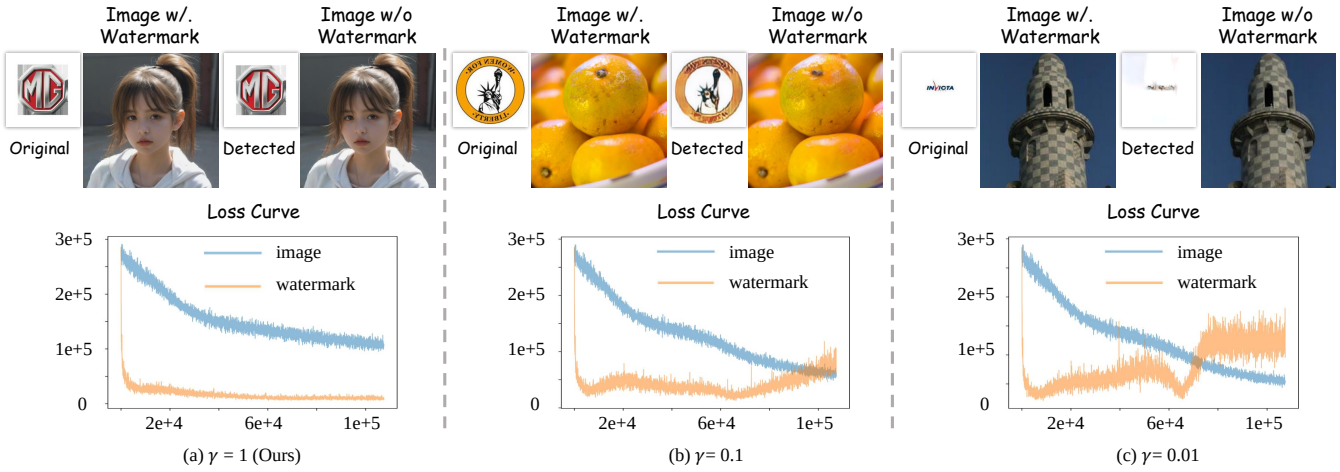
The time period of λ-encryption. To further explore the influence of different injection time of watermark on image synthesis quality, we also perform a study on the watermarking time period t, as shown in Figure 7. From Figure 7, we can observe that the earlier the injection occurs, the less high-frequency information in the image is retained in the final generative results. Particularly, when $\lambda = 15$ and the watermarking time period is in the early stage (refer to the first column of each case in Figure 7(c)), it will cause image distortion, which indicates that the watermarking unit (Figure 3) should be activated set as often as possible during the middle to end time period of latent diffusion, for better balancing watermark injection and generative effects.

4.4 Analysis on hyper-parameter γ

To further trading off the high-fidelity image synthesis and high-traceable watermark injection, we perform this study on hyper-parameter γ (refer to Formula 3), as illustrated in Figure 8. From Figure 8(a), we can observe that: **1)** When the loss of image reconstruction and the loss of watermark decoding have the same weight (*i.e.*, $\gamma = 1$), both of them can steadily decrease until the model converges; **2)** When reducing γ to make the model focus on image synthesis (*i.e.*, $\gamma = 0.1$), the loss curves of both have a significant decline in the early stage, but after that the watermark decoding becomes difficult to converge. Correspondingly, the decoded LOGO has become obviously blurred at this time; **3)** When γ further decreases (*i.e.*, $\gamma = 0.01$), similar conclusion is further verified. Based on the above discussion, we finally choose $\gamma = 1$ to balance image synthesis and watermark decoding.

4.5 The robustness of watermarking

Anti-attack test. We conduct the anti-attack test to evaluate the robustness of our graphical watermarking against a variety



(a) γ = 1 (Ours) (b) γ= 0.1 (c) γ= 0.01

Figure 8: The effect of the hyper-parameter γ. The generated images, watermarks and the curve of loss value are shown to qualitatively and quantitatively assess the effect of γ.

	PSNR ↑	FID ↓	LPIPS ↓	CLIP-Score ↑
None (Ours)	**33.17**	**18.89**	**0.232**	**88.15**
Rotate 90	32.96	18.94	0.228	87.72
Resize 0.7	32.18	19.11	0.242	87.03
Brightness 2.0	30.53	19.77	0.257	86.09
Crop 10%	31.01	19.30	0.250	85.01
Combined	29.24	20.18	0.275	83.49

Table 3: Robustness studies on LSUN-Churches dataset.

of attacks, as shown in Table 3. Specifically, we follow Stable Signature [15] to set up 5 attack tests: **1)** *Rotate 90*, **2)** *Resize 0.7*, **3)** *Brightness 2.0*, **4)** *Crop 10%*, **5)** *Combined*. From Table 3, we can observe that our approach is robust to the various attacks. For examples, the PSNR metric under the most challenging combined attack is still higher than 29%, and the LPIPS metric is still lower than 0.28%, which demonstrate the excellent robustness of our Safe-SD. Moreover, the CLIP-Scores under all attacks are still higher than 83%, which demonstrate most of the semantic information is still retained in the watermarked images. Moreover, it can be observed that the brightness has relatively maximal impact on generation quality (*e.g.*, PSNR, FID, LPIPS), and even if the image is cropped to 10% of the original image, it still retains a high watermark recognition rate, which verifies the effectiveness of Safe-SD.

Multi-watermarking test. Figure 9 shows the test results of multiple watermarking. From Figure 9, we can notice that when multiple watermarks are injected at the same time, our Safe-SD still could maintain the high-quality image characteristics. Meanwhile, the two injected watermarks in Figure 9 can still be clearly extracted, demonstrating the superiority of our model in multi-watermarking scenarios.

5 CONCLUSION

In this paper, we have presented Safe-SD, a safe and high-traceable Stable Diffusion framework with text prompt trigger for unified

Figure 9: Multiple watermarking evaluations.

generative watermarking and detection. Specifically, we design a simple and unified architecture, which makes it possible to simultaneously train watermark injection and detection in a single network, greatly improving the efficiency and convenience of use. Moreover, to further support text-driven generative watermarking, we elaborately design a λ-sampling and λ-encryption algorithm to fine-tune a latent diffuser wrapped by a VAE for balancing high-fidelity image synthesis and high-traceable watermark detection. Besides, we introduce a novel prompt triggering mechanism to enable adaptive watermark injection for facilitating copyright protection. Note the proposed approach can be easily extended to other diffusion models and can adapt to various downstream tasks. Experiments on the representative LSUN-Churches, COCO, and FFHQ datasets demonstrate the effectiveness and superior performance of our Safe-SD model in both quantitative and qualitative evaluations.

REFERENCES

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

[2] Shumeet Baluja. 2019. Hiding images within images. *IEEE transactions on pattern analysis and machine intelligence* 42, 7 (2019), 1685–1697.

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18392–18402.

[4] Huili Chen, Bita Darvish Rouhani, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. 2019. Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*. 105–113.

[5] Huili Chen, Bita Darvish Rouhani, and Farinaz Koushanfar. 2019. Blackmarks: Blackbox multibit watermarking for deep neural networks. *arXiv preprint arXiv:1904.00344* (2019).

[6] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526* (2017).

[7] Betty Cortiñas-Lorenzo and Fernando Pérez-González. 2020. Adam and the Ants: On the Influence of the Optimization Algorithm on the Detectability of DNN Watermarks. *Entropy* 22, 12 (2020), 1379.

[8] Ingemar Cox, Matthew Miller, Jeffrey Bloom, and Chris Honsinger. 2002. Digital watermarking. *Journal of Electronic Imaging* 11, 3 (2002), 414–414.

[9] Yingqian Cui, Jie Ren, Han Xu, Pengfei He, Hui Liu, Lichao Sun, and Jiliang Tang. 2023. DiffusionShield: A Watermark for Copyright Protection against Generative Diffusion Models. *arXiv preprint arXiv:2306.04642* (2023).

[10] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. 2019. Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. 485–497.

[11] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in NeurIPS* 34 (2021), 8780–8794.

[12] Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *CVPR*. 12873–12883.

[13] Lixin Fan, Kam Woh Ng, and Chee Seng Chan. 2019. Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks. *Advances in neural information processing systems* 32 (2019).

[14] Jianwei Fei, Zhihua Xia, Benedetta Tondi, and Mauro Barni. 2022. Supervised gan watermarking for intellectual property protection. In *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*. 1–6.

[15] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. 2023. The stable signature: Rooting watermarks in latent diffusion models. *arXiv preprint arXiv:2303.15435* (2023).

[16] Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. 2022. Watermarking images in self-supervised latent spaces. In *ICASSP*. IEEE, 3054–3058.

[17] Matthew Gault. 2022. An AI-Generated Artwork Won First Place at a State Fair Fine Arts Competition, and Artists Are Pissed. *URL: https://www.vice.com/en/article/bvmvqm/an-aigenerated-artwork-won-first-place-at-a-state-fair-fine-arts-competition-and-artists-are-pissed* (2022).

[18] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. 2022. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*. 10696–10706.

[19] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access* 7 (2019), 47230–47244.

[20] Jia Guo and Miodrag Potkonjak. 2019. Evolutionary trigger set generation for dnn black-box watermarking. *arXiv preprint arXiv:1906.04411* (2019).

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in NeurIPS* 33 (2020), 6840–6851.

[22] Jonathan Ho and Tim Salimans. 2021. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop*.

[23] Hengrui Jia, Christopher A Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot. 2021. Entangled watermarks as a defense against model extraction. In *30th USENIX Security Symposium (USENIX Security 21)*. 1937–1954.

[24] Zhengyuan Jiang, Jinghuai Zhang, and Neil Zhenqiang Gong. 2023. Evading Watermark based Detection of AI-Generated Content. *arXiv preprint arXiv:2305.03807* (2023).

[25] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4401–4410.

[26] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. 2022. Denoising diffusion restoration models. *arXiv preprint arXiv:2201.11793* (2022).

[27] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. 2023. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6007–6017.

[28] Dongjun Kim, Byeonghu Na, Se Jung Kwon, Dongsoo Lee, Wanmo Kang, and Il-Chul Moon. 2022. Maximum Likelihood Training of Implicit Nonlinear Diffusion Models. *arXiv preprint arXiv:2205.13699* (2022).

[29] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. 2021. Variational diffusion models. *Advances in NeurIPS* 34 (2021), 21696–21707.

[30] Varsha Kishore, Xiangyu Chen, Yan Wang, Boyi Li, and Kilian Q Weinberger. 2021. Fixed neural network steganography: Train the images, not the network. In *ICLR*.

[31] Hyun Kwon and Yongchul Kim. 2022. BlindNet backdoor: Attack on deep neural network using blind watermark. *Multimedia Tools and Applications* (2022), 1–18.

[32] Huiying Li, Emily Willson, Haitao Zheng, and Ben Y Zhao. [n. d.]. Persistent and unforgeable watermarks for deep neural networks. *arXiv preprint arXiv:1910.01226* ([n. d.]).

[33] Yue Li, Benedetta Tondi, and Mauro Barni. 2021. Spread-transform dither modulation watermarking of deep neural network. *Journal of Information Security and Applications* 63 (2021), 103004.

[34] Yue Li, Hongxia Wang, and Mauro Barni. 2021. A survey of deep neural network watermarking techniques. *Neurocomputing* 461 (2021), 171–193.

[35] Zheng Li, Chengyu Hu, Yang Zhang, and Shanqing Guo. 2019. How to prove your model belongs to you: A blind-watermark based framework to protect intellectual property of DNN. In *Proceedings of the 35th Annual Computer Security Applications Conference*. 126–137.

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *roceedings of the European conference on computer vision (ECCV)*. 740–755.

[37] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. 2022. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778* (2022).

[38] Xihui Liu, Dong Huk Park, Samaneh Azadi, Gong Zhang, Arman Chopikyan, Yuxiao Hu, Humphrey Shi, Anna Rohrbach, and Trevor Darrell. 2021. More control for free! image synthesis with semantic diffusion guidance. *arXiv preprint arXiv:2112.05744* (2021).

[39] Nils Lukas, Yuxuan Zhang, and Florian Kerschbaum. 2020. Deep Neural Network Fingerprinting by Conferrable Adversarial Examples. In *International Conference on Learning Representations*.

[40] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2021. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *International Conference on Learning Representations*.

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

[42] Ryota Namba and Jun Sakuma. 2019. Robust watermarking of neural network with exponential weighting. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*. 228–240.

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

[44] Alexander Quinn Nichol and Prafulla Dhariwal. 2021. Improved denoising diffusion probabilistic models. In *ICML*. 8162–8171.

[45] Ding Sheng Ong, Chee Seng Chan, Kam Woh Ng, Lixin Fan, and Qiang Yang. 2021. Protecting intellectual property of generative adversarial networks from ambiguity attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3630–3639.

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*. 8748–8763.

[47] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).

[48] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*. 10684–10695.

[49] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22500–22510.

[50] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *arXiv preprint arXiv:2205.11487* (2022).

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

[52] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. 2021. Maximum likelihood training of score-based diffusion models. *Advances in NeurIPS* 34 (2021), 1415–1428.

[53] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* (2020).

[54] Sebastian Szyller, Buse Gul Atli, Samuel Marchal, and N Asokan. 2021. Dawn: Dynamic adversarial watermarking of neural networks. In *Proceedings of the 29th ACM International Conference on Multimedia*. 4417–4425.

[55] Enzo Tartaglione, Marco Grangetto, Davide Cavagnino, and Marco Botta. 2021. Delving in the loss landscape to embed robust watermarks into neural networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*. 1243–1250.

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

[57] Arash Vahdat, Karsten Kreis, and Jan Kautz. 2021. Score-based generative modeling in latent space. *Advances in NeurIPS* 34 (2021), 11287–11302.

[58] Jing Wang, Weiqing Min, Sujuan Hou, Shengnan Ma, Yuanjie Zheng, Haishuai Wang, and Shuqiang Jiang. 2020. Logo-2K+: A large-scale logo dataset for scalable logo classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6194–6201.

[59] Jiangfeng Wang, Hanzhou Wu, Xinpeng Zhang, and Yuwei Yao. 2020. Watermarking in deep neural networks via error back-propagation. *Electronic Imaging* 2020, 4 (2020), 22–1.

[60] Tianhao Wang and Florian Kerschbaum. 2019. Attacks on digital watermarks for deep neural networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2622–2626.

[61] Tianhao Wang and Florian Kerschbaum. 2019. Robust and undetectable white-box watermarks for deep neural networks. *arXiv preprint arXiv:1910.14268* 1, 2 (2019).

[62] Hanzhou Wu, Gen Liu, Yuwei Yao, and Xinpeng Zhang. 2020. Watermarking neural networks with watermarked images. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 7 (2020), 2591–2601.

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

[64] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. 2022. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789* (2022).

[65] Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. 2021. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International conference on computer vision*. 14448–14457.

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

[67] Lvmin Zhang and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543* (2023).

[68] Xiangyu Zhao, Hanzhou Wu, and Xinpeng Zhang. 2021. Watermarking graph neural networks by random graphs. In *2021 9th International Symposium on Digital Forensics and Security (ISDFS)*. 1–6.

[69] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai-Man Cheung, and Min Lin. 2023. A recipe for watermarking diffusion models. *arXiv preprint arXiv:2303.10137* (2023).

[70] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. 2018. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*. 657–672.