Gaussian Shading: Provable Performance-Lossless Image Watermarking for Diffusion Models

Zijin Yang[1], Kai Zeng[1], Kejiang Chen[1,*], Han Fang[2], Weiming Zhang[1], Nenghai Yu[1],
[1]Anhui Province Key Laboratory of Digital Security, University of Science and Technology of China
[2]National University of Singapore
{bsmhmmlf@mail., zk0128@mail., chenkj@, zhangwm@, ynh@}ustc.edu.cn fanghan@nus.edu.sg

Abstract

Ethical concerns surrounding copyright protection and inappropriate content generation pose challenges for the practical implementation of diffusion models. One effective solution involves watermarking the generated images. However, existing methods often compromise the model performance or require additional training, which is undesirable for operators and users. To address this issue, we propose Gaussian Shading, a diffusion model watermarking technique that is both performance-lossless and training-free, while serving the dual purpose of copyright protection and tracing of offending content. Our watermark embedding is free of model parameter modifications and thus is plug-and-play. We map the watermark to latent representations following a standard Gaussian distribution, which is indistinguishable from latent representations obtained from the non-watermarked diffusion model. Therefore we can achieve watermark embedding with lossless performance, for which we also provide theoretical proof. Furthermore, since the watermark is intricately linked with image semantics, it exhibits resilience to lossy processing and erasure attempts. The watermark can be extracted by Denoising Diffusion Implicit Models (DDIM) inversion and inverse sampling. We evaluate Gaussian Shading on multiple versions of Stable Diffusion, and the results demonstrate that Gaussian Shading not only is performance-lossless but also outperforms existing methods in terms of robustness.

1. Introduction

Diffusion models [16, 31–34] signify a noteworthy leap forward in image generation. These well-trained diffusion models, especially commercial diffusion models like Stable Diffusion (SD) [30], Glide [27], and Muse AI [30], enable individuals with diverse backgrounds to create high-quality images effortlessly. However, this raises concerns about intellectual property and whether diffusion models will be stolen or resold twice.

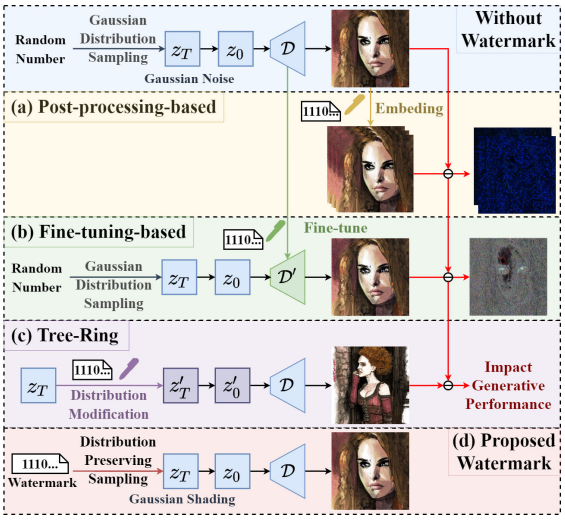


Figure 1. Existing watermarking frameworks can be divided into three categories: post-processing-based, fine-tuning-based, and latent-representation-based Tree-Ring. Our method also relies on latent representations but achieves performance-lossless without altering the distribution.

On the other hand, the ease of generating realistic images raises concerns about potentially misleading content generation. For example, on May 23, 2023, a Twitter-verified user named Bloomberg Feed posted a tweet titled "Large explosion near the Pentagon complex in Washington DC-initial report," along with a synthetic image. This tweet led to multiple authoritative media accounts sharing it, even causing a brief impact on the stock market[1]. On October 30, 2023, White House issued an executive order on AI security, emphasizing the need to protect Americans from AI-enabled fraud and deception by establishing standards and best practices for detecting AI-generated content and authenticating official content[2]. The urgency of labeling gen-

*Corresponding author

[1]Fake image of Pentagon explosion on Twitter

[2]FACT SHEET: President Biden Issues Executive Order on Safe, Secure, and Trustworthy Artificial Intelligence

erated content for copyright authentication and prevention of misuse is evident.

Watermarking is highlighted as a fundamental method for labeling generated content, as it embeds watermark information within the generated image, allowing for subsequent copyright authentication and the tracking of false content. Existing watermarking methods for the diffusion model can be divided into three categories, as shown in Fig. 1. Post-processing-based watermarks [6, 45] adjust robust image features to embed watermarks, thereby directly altering the image and degrading its quality. To mitigate this concern, recent research endeavors propose fine-tuning-based methods [7, 10, 24, 42, 49], which amalgamate the watermark embedding process with the image generation process. Intuitively, these methods need to modify model parameters, introducing supplementary computational overhead. Recently, Wen et al. [41] proposed the latent-representation-based Tree-Ring watermark, which conveys information by adapting the latent representations to match specific patterns. However, it restricts the randomness of sampling, which impacts generative performance.

Through the above analysis, we can find that these methods compromise model performance to embed watermarks. In practical applications, model performance is paramount for both business interests and user experience. Substantial resource investment is often necessary to pursue enhanced model performance. This leads to a fundamental question: Can watermarks be embedded without compromising model performance?

We affirmatively address the question presented above. Succinctly, the generation process can be delineated into two key phases: latent representation sampling and decoding. Our goal is to align the distribution of the latent representation in watermarked images with that of the latent representation in normally generated images. By keeping the model unaltered, the distribution of watermarked images is naturally consistent with that of normally generated images, enabling the seamless embedding of watermarks without compromising model performance.

Building upon this insight, we propose a watermarking method named Gaussian Shading, designed to ensure no deterioration in model performance. The embedding process encompasses three primary elements: watermark diffuse, randomization, and distribution-preserving sampling. Watermark diffusion spreads the watermark information throughout the latent representation. During the generation process, the watermark information will be diffused to the whole semantics of the image, thus achieving excellent robustness. Watermark randomization and distribution-preserving sampling guarantee the congruity of the latent representation distribution with that of watermark-free latent representations, thereby achieving performance-lossless. In the extraction phase, the latent representations are acquired through Denoising Diffusion Implicit Model (DDIM) inversion [32], allowing for the retrieval of watermark information. Harnessing the extensive scope of the SD latent space, we can achieve a high-capacity watermark of 256 bits, surpassing prior methods.

To the best of our knowledge, ours is the first technique that tackles this challenging problem of performance-lossless watermarking for diffusion models, and we provide theoretical proof. Moreover, this technique leaves the architecture and parameters of SD unaltered, necessitating no supplementary training. It can seamlessly integrate as a plug-and-play module within the generation process. Model providers can easily replace watermarked models with non-watermarked ones without affecting usability experiences.

We conducted thorough experiments on SD. Under strong noise perturbation, the average true positive rate and bit accuracy can exceed 0.99 and 0.97, respectively, validating the superiority of Gaussian Shading in both detection and traceability tasks compared to prior methods. Additionally, experiments on visual quality and image-text similarity serve as indicators of performance preservation in our approach. Lastly, we deliberated on various watermark erasure attacks, affirming the steadfast performance of our watermark in the face of such adversities.

2. Related Work

2.1. Diffusion Models

Inspired by non-equilibrium thermodynamics, Ho et al. [16] introduced the Denoising Diffusion Probabilistic Model (DDPM). DDPM consists of two Markov chains used for adding and removing noise, and subsequent works [8, 11, 15, 25, 28, 30, 32] have adopted this bidirectional chain framework. To reduce computational complexity and improve efficiency, the Latent Diffusion Model (LDM) [30] was designed, in which the diffusion process occurs in a latent space $\mathcal{Z}$. To map an image $x \in \mathbb{R}^{H\times W\times 3}$ to the latent space, the LDM employs an encoder $\mathcal{E}$, such that $z_0 = \mathcal{E}(x) \in \mathbb{R}^{h\times w\times c}$. Similarly, to reconstruct an image from the latent space, a decoder $\mathcal{D}$ is used, such that $x = \mathcal{D}(z_0)$. A pretrained LDM can generate images without the encoder $\mathcal{E}$. Specifically, a latent representation z_T is first sampled from a standard Gaussian distribution $\mathcal{N}(0, I)$. Subsequently, through iterative denoising using methods like DDIM [32], z_0 is obtained, and an image can be generated using the decoder: $x = \mathcal{D}(z_0)$.

2.2. Image Watermarking

Digital watermarking [38] is an effective means to address copyright protection and content authentication by embedding copyright or traceable identification information within carrier data. Typically, the functionality of a watermark depends on its capacity. For example, a single-bit

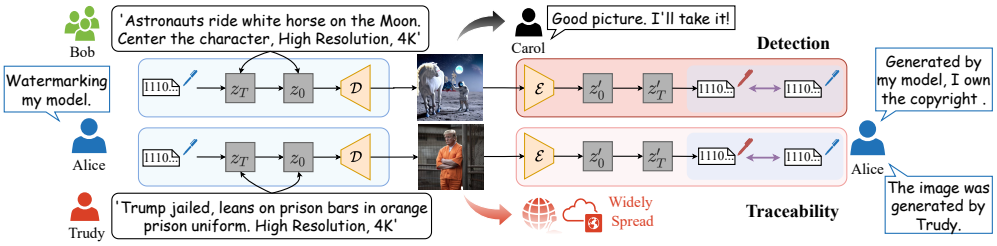


Figure 2. Application scenarios for Gaussian Shading.

watermark can determine whether an image was generated by a particular diffusion model, i.e., copyright protection; a multi-bit watermark can further determine which user of the diffusion model generated the image, i.e., traceability.

Image watermarking is a method that employs images as carriers for watermarking. Initially, watermark embedding methods primarily focused on the spatial domain [38], but later, to enhance robustness, transform domain watermarking techniques [1, 12, 13, 20, 22, 35, 37] were developed. In recent years, with the advancement of deep learning, researchers have turned their attention to neural networks [21, 39], harnessing their powerful learning capabilities to develop watermarking techniques [18, 19, 26, 36, 43, 50, 51].

2.3. Image Watermarking for Diffusion Models

Existing Image watermarking methods for the diffusion model [6, 7, 10, 24, 41, 42, 45, 49] can be divided into three categories, as shown in Fig. 1. The image watermarking methods described in the previous section can be applied directly to the images generated by the diffusion model, which is called post-processing-based watermarks [6, 45]. These methods directly modify the image, thus degrading image quality. Recent research endeavors have amalgamated the watermark embedding process with the image generation process to mitigate this issue. Stable Signature [10] fine-tunes the LDM decoder using a pre-trained watermark extractor, facilitating watermark extraction from images produced by the fine-tuned model. Zhao et al. [49] and Liu et al. [24] suggest fine-tuning the diffusion model to implant a backdoor as a watermark, enabling watermark extraction by triggering. These fine-tuning-based approaches enhance the quality of watermarked images but introduce supplementary computational overhead and modify model parameters. Furthermore, Wen et al. [41] introduced the Tree-Ring Watermark, which conveys copyright information by adapting the frequency domain of latent representations to match specific patterns. This method achieves an imperceptible watermark. However, it directly disrupts the Gaussian distribution of noise, limiting the randomness of sampling and resulting in affecting model performance.

3. Methods

In this section, we provide an overview of the application scenarios and functionalities in Fig. 2. We then proceed to detail the embedding and extraction processes shown in Fig. 3. Finally, we present a mathematical proof of the performance-lossless characteristic of the watermark.

3.1. Application Scenarios

Scenarios. See Fig. 2, the scenario involves the operator Alice, the thief Carol, and two types of users Bob and Trudy.

Alice is responsible for training the model, deploying it on the platform, and providing the corresponding API for users, but she does not open-source the code or model weights. Carol does not use Alice's services but steals images generated by her model, claiming ownership of the copyrights. Bob and Trudy, as community users, can utilize the API to generate and disseminate images. While Bob faithfully adheres to the community guidelines, Trudy aims to generate deep fake, and infringing content. To evade detection and traceability, Trudy can employ various data augmentation to modify illicit images.

Detection. This scenario satisfies the detection (copyright protection) requirement. Alice embeds a single-bit watermark into each generated image. The successful extraction of the watermark from an image serves as evidence of Alice's rightful ownership of the copyright, while also indicating that the image is artificially generated (as opposed to natural images).

Traceability. This scenario fulfills the traceability requirement. Alice allocates a watermark to each user. By extracting the watermark from the illicit content, it enables tracing Trudy, through comparison with the watermark database. Traceability is a higher pursuit than detection and can also achieve copyright protection for different users.

Details of the statistical tests in both scenarios are shown in Supplementary Material.

3.2. Watermark Embedding

Watermark diffusion. The dimensions of the latent representations are given by $c \times h \times w$, where each dimension can represent l bits of the watermark. Therefore, the watermark capacity becomes $l \times c \times h \times w$ bits. To enhance the robustness of the watermark, we represent the watermark using $\frac{1}{f_{hw}}$ of the height and width, and $\frac{1}{f_c}$ of the channel, and replicate the watermark $f_c \cdot f_{hw}^2$ times. Thus, the watermark s with dimensions $l \times \frac{c}{f_c} \times \frac{h}{f_{hw}} \times \frac{w}{f_{hw}}$ is expanded into a diffused watermark s^d with dimensions $l \times c \times h \times w$.

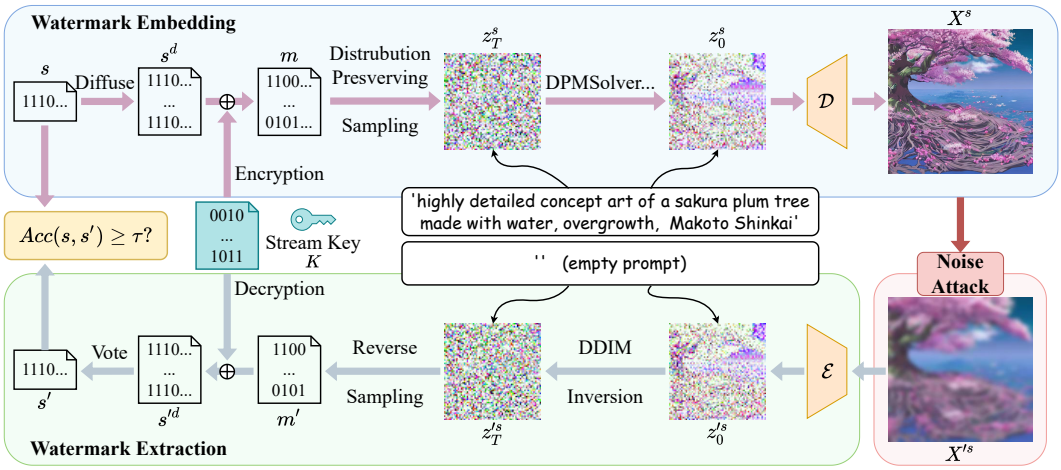


Figure 3. The framework of Gaussian Shading. We utilize a k-bit binary sequence s to represent the watermark. After diffusion and encryption, the watermark can be utilized to drive distribution-preserving sampling, followed by denoising to generate watermarked images X^s. For extraction, it is sufficient to introduce DDIM inversion and the inverse process of all the operations mentioned above.

The actual watermark capacity is $k = \frac{l \times c \times h \times w}{f_c \cdot f_{hw}^2}$ bits.

Watermark randomization. If we know the distribution of the diffused watermark s^d, we can directly utilize distribution-preserving sampling to obtain the corresponding latent representations z_T^s. However, in practical scenarios, its distribution is always unknown. Hence, we introduce a stream key K to transform s^d into a distribution-known randomized watermark m through encryption. Considering the use of computationally secure stream cipher, such as ChaCha20 [3], m follows a uniform distribution, i.e., m is a random binary bit stream.

Distribution-preserving sampling driven by randomized watermark. When each dimension represents l-bit randomized watermark m, this l bits can be regarded as an integer $y \in [0, 2^l - 1]$. Since m is a ciphertext, y follows a discrete uniform distribution, i.e., $p(y) = \frac{1}{2^l}$ for $y = 0, 1, 2, \ldots, 2^l - 1$. Let $f(x)$ denote the probability density function of the Gaussian distribution $\mathcal{N}(0, I)$, and ppf denotes the quantile function. We divide $f(x)$ into 2^l equal cumulative probability portions. When $y = i$, the watermarked latent representation z_T^s falls into the i-th interval, which means z_T^s should follow the conditional distribution:

$$p(z_T^s|y=i) = \begin{cases} 2^l \cdot f(z_T^s) & ppf(\frac{i}{2^l}) < z_T^s \leq ppf(\frac{i+1}{2^l}) \\ 0 & otherwise \end{cases}. \tag{1}$$

The probability distribution of z_T^s is given by:

$$p(z_T^s) = \sum_{i=0}^{2^l-1} p(z_T^s|y=i)p(y=i) = f(z_T^s). \tag{2}$$

Eq. (2) indicates that z_T^s follows the same distribution as the randomly sampled latent representation $z_T \sim \mathcal{N}(0, I)$. Next, we elaborate on how this sampling is implemented.

Let the cumulative distribution function of $f(x)$ be denoted as cdf. We can obtain the cumulative distribution function of Eq. (1) as follows,

$$F(z_T^s|y=i) = \begin{cases} 0 & z_T^s < ppf(\frac{i}{2^l}) \\ 2^l \cdot cdf(z_T^s) - i & ppf(\frac{i}{2^l}) \leq z_T^s \leq ppf(\frac{i+1}{2^l}) \\ 1 & z_T^s > ppf(\frac{i+1}{2^l}) \end{cases}. \tag{3}$$

Given $y = i$, we aim to perform random sampling of z_T^s within the interval $[ppf(\frac{i}{2^l}), ppf(\frac{i+1}{2^l})]$. The commonly used method is rejection sampling [4, 17, 47], which can be time-consuming as it requires repeated sampling until z_T^s falls into the correct interval. Instead, we can utilize the cumulative probability density. When randomly sampling $F(z_T^s|y=i)$, the corresponding z_T^s is naturally obtained through random sampling. Since $F(z_T^s|y=i)$ takes values in $[0, 1]$, sampling from it is equivalent to sampling from a standard uniform distribution, denoted as $u = F(z_T^s|y=i) \sim \mathcal{U}(0, 1)$. Shift the terms of Eq. (3), and take into account that cdf and ppf are inverse functions, we have

$$z_T^s = ppf(\frac{u+i}{2^l}). \tag{4}$$

Eq. (4) represents the process of sampling the watermarked latent representation z_T^s driven by the randomized watermark m. To extract the watermark, its inverse map is

$$i = \lfloor 2^l \cdot cdf(z_T^s) \rfloor. \tag{5}$$

Image generation. After the sampling process, the watermark is embedded in the latent representation z_T^s, and the subsequent generation process is no different from the regular generation process of SD. Here, we employ the DPM-Solver [25] for iterative denoising of z_T^s, In addition to DPMSolver [25], other continuous-time samplers based on ordinary differential equation (ODE) solvers [32], such as DDIM [32], DEIS [46], PNDM [23], and UniPC [48], can be used too. After obtaining denoised z_0^s, the watermarked image X^s is generated using the decoder $\mathcal{D}$: $X^s = \mathcal{D}(z_0^s)$.

3.3. Watermark Extraction

DDIM Inversion. Using the SD encoder $\mathcal{E}$, we first restore X'^s to the latent space $z'^s_0 = \mathcal{E}(X'^s)$. Then, we introduce the DDIM inversion [32] to estimate the additive noise. It can be considered that $z'^s_T \approx z^s_T$. We also observe that although DDIM inversion is derived from DDIM, it can apply to other continuous-time samplers based on ODE solvers.

Watermark reduction from latent representations. After obtaining z'^s_T, according to the inverse transformation defined in Eq. (5), the tensor can be converted into a bit stream m'. Subsequently, m' is decrypted using K to obtain s'^d. Inverse diffusion of the watermark results in $f_c \cdot f^2_{hw}$ copies of the watermark. Similar to voting, if the bit is set to 1 in more than half of the copies, the corresponding watermark bit is set to 1; otherwise, it is set to 0. This process restores the true binary watermark sequence s'.

3.4. Proof of Lossless Performance

In prior works, the incorporation of watermark embedding modules inevitably results in a decline in model performance, as typically evaluated using metrics such as Peak Signal-to-Noise Ratio (PSNR) and Fréchet Inception Distance (FID) [14], which are more suitable for assessing post-processing methods. To assess methods that integrate the watermark embedding and generation processes, we propose a definition for the impact of watermark embedding on model performance, drawing on the complexity-theoretic definition of steganographic security [17]. This definition is based on a probabilistic game between a watermarked image X^s and a normally generated image X. The tester $\mathcal{A}$ can use any watermark to drive the sampling process and generate X^s, similar to the *chosen hidden text attacks* [17], which we refer to as *chosen watermark tests*. The watermarking method is performance-lossless under *chosen watermark tests*, if for any polynomial-time tester $\mathcal{A}$ and key $K \leftarrow \mathsf{KeyGen}_{\mathcal{G}(1^\rho)}$, it holds that

$$\left|\Pr\left[\mathcal{A}\left(X^s\right)=1\right]-\Pr\left[\mathcal{A}\left(X\right)=1\right]\right|<\mathsf{negl}\left(\rho\right). \quad (6)$$

Here, ρ represents the length of the security parameter, such as the key K, and $\mathsf{negl}(\rho)$ is a negligible term relative to ρ.

We prove the statement using a proof by contradiction. First, assume that the watermarked image X^s and the normally generated image X are distinguishable, meaning

$$\left|\Pr\left[\mathcal{A}\left(X^s\right)=1\right]-\Pr\left[\mathcal{A}\left(X\right)=1\right]\right|=\delta, \quad (7)$$

where δ is non-negligible with respect to the key K. Let the iterative denoising process be denoted as $Q(\cdot)$, and substitute the LDM encoder $\mathcal{E}$ into Eq. (7), we have

$$\begin{aligned}&\left|\Pr\left[\mathcal{A}\left(\mathcal{E}\left(Q\left(z^s_T\right)\right)\right)=1|m=E\left(K,s^d\right)\right]\right.\\&\left.-\Pr\left[\mathcal{A}\left(\mathcal{E}\left(Q\left(z_T\right)\right)\right)=1|z_T\leftarrow\mathcal{N}\left(0,I\right)\right]\right|=\delta,\end{aligned} \quad (8)$$

where the randomized watermark m is obtained by encrypting the diffused watermark s^d using the encryption algorithm E with key K. Note that Eq. (2) contains the fact that distribution-preserving sampling driven by randomized watermark and random sampling are equivalent. Therefore, we denote sequence-driven sampling as $S(\cdot)$. z^s_T can naturally be obtained by sampling driven by m, i.e., $z^s_T = S(m)$. On the other hand, z_T can be considered as obtained by sampling driven by a truly random sequence r of the same length as m, i.e., $z_T = S(r)$. Eq. (8) can be written as

$$\begin{aligned}&\left|\Pr[\mathcal{A}\left(\mathcal{E}\left(Q\left(S\left(m\right)\right)\right)\right)=1|m=E(K,s^d)]\right.\\&\left.-\Pr\left[\mathcal{A}\left(\mathcal{E}\left(Q\left(S\left(r\right)\right)\right)\right)=1\right]\right|=\delta.\end{aligned} \quad (9)$$

Sampling $S(\cdot)$, denoising $Q(\cdot)$, and encoder $\mathcal{E}$ can be considered as subroutines that the tester $\mathcal{A}_{\mathcal{E},Q,S}$ can use. Thus, Eq. (9) can be simplified,

$$\begin{aligned}&\left|\Pr[\mathcal{A}_{\mathcal{E},Q,S}\left(m\right)=1|m=E\left(K,s^d\right)]\right.\\&\left.-\Pr\left[\mathcal{A}_{\mathcal{E},Q,S}\left(r\right)=1\right]\right|=\delta.\end{aligned} \quad (10)$$

Note that $S(\cdot)$, $Q(\cdot)$, and $\mathcal{E}$ are all polynomial-time programs, so the time taken by the tester $\mathcal{A}_{\mathcal{E},Q,S}$ to make the distinction is also polynomial. Eq. (10) essentially states that it is possible to distinguish between m and r in polynomial time. However, we have used the computationally secure stream cipher ChaCha20 [3] in watermark randomization, which means that m as a pseudorandom sequence cannot be distinguished from a truly random sequence in polynomial time. Eq. (10) contradicts the computational security property of ChaCha20 [3]. Therefore, Eq. (10) is not valid, leading us back to our initial assumption that Eq. (7) is also not valid. This implies that the watermarked image X^s and the normally generated image X are indistinguishable in polynomial time. Hence, Gaussian Shading is performance-lossless under *chosen watermark tests*.

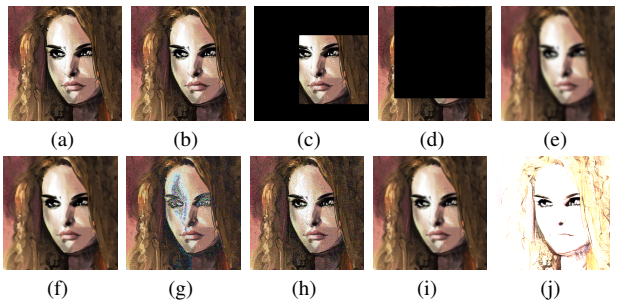
4. Experiments

This section focuses on experimental analysis, including details of the experimental setup, performance evaluation of Gaussian Shading, comparison with baseline methods, ablation experiments, and potential attacks.

4.1. Implementation Details

SD models. In this paper, we focus on text-to-image LDM, hence we select SD [30] provided by huggingface. We evaluate Gaussian Shading as well as baseline methods, using three versions of SD: V1.4, V2.0, and V2.1. The size of the generated images is 512×512, and the latent space dimension is $4 \times 64 \times 64$. During inference, we employ the prompt from Stable-Diffusion-Prompt[3], with a guidance scale of 7.5. We sample 50 steps using DPMSolver [25]. Considering that users tend to propagate the generated images without retaining the corresponding prompts, we use an empty prompt for inversion, with a scale of 1. We perform 50 steps of inversion using DDIM inversion [32].

[3] Stable-Diffusion-Prompts



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 4. Watermarked image is attacked by different noise. (a) Watermarked image. (b) JPEG, $QF = 25$. (c) 60% area Random Crop (RandCr). (d) 80% area Random Drop (RandDr). (e) Gaussian Blur, $r = 4$ (GauBlur). (f) Median Filter, $k = 7$ (MedFilter). (g) Gaussian Noise, $\mu = 0$, $\sigma = 0.05$ (GauNoise). (h) Salt and Pepper Noise, $p = 0.05$ (S&PNoise). (i) 25% Resize and restore (Resize). (j) Brightness, $factor = 6$.

Watermarking methods. In the main experiments, the settings for Gaussian Shading are $f_c = 1, f_{hw} = 8, l = 1$, resulting in an actual capacity of 256 bits. We select five baseline methods: three officially used by SD, namely DwtDct [6], DwtDctSvd [6], and RivaGAN [45], a multi-bit watermarking called Stable Signature [10], and a train-free invisible watermarking called Tree-Ring [41].

Robustness evaluation To evaluate the robustness, we select nine representative types of noise shown in Fig. 4. We conduct experiments following the noise strength in Fig. 4.

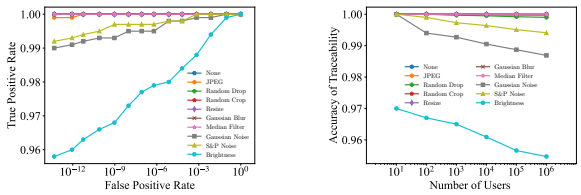
Evaluation metrics. In the detection scenario, we calculate the true positive rate (TPR) corresponding to a fixed false positive rate (FPR). In the traceability scenario, we calculate the bit accuracy. To measure the bias in model performance, we compute the FID [14] and CLIP-Score [29] for 10 batches of watermarked images and perform a t-test on the mean FID and CLIP-Score compared to that of watermark-free images.

All experiments are conducted using the PyTorch 1.13.0 framework, running on a single RTX 3090 GPU.

4.2. Performance of Gaussian Shading

Detection. In the detection scenario, we consider Gaussian Shading as a single-bit watermark, with a fixed watermark s. We approximate the FPR to be controlled at $10^0, 10^{-1}, \ldots, 10^{-13}$, calculate the corresponding threshold τ , and test the TPR on $1,000$ watermarked images To mitigate the effects of randomness, we perform 5 trials with different s and compute the average TPR. See Fig. 5a, when the FPR is controlled at 10^{-13}, the TPR remains at least 0.99 for eight out of the nine cases. Although the TPR for Brightness is only 0.953, it is still a promising result.

Traceability. In this scenario, Gaussian Shading serves as a multi-bit watermark. Assuming Alice provides services to N users, Alice needs to allocate one watermark for each user. In our experiments, we assume that $N' = 1,000$ users generate images, with each user generating 10 images, resulting in a dataset of $10,000$ watermarked images.



(a) Detection results. (b) Traceability results.

Figure 5. Performance of Gaussian Shading.

During testing, we calculate the threshold τ to control the FPR at 10^{-6}. Note that when computing traceability accuracy, we need to consider two types of errors: false positives, where watermarked images are not detected, and traceability errors, where watermarked images are detected but attributed to the wrong user. Therefore, we first determine whether the image contains a watermark. If it does, we calculate the number of matching bits Acc with all N users on the platform. The user with the highest Acc is considered the one who generated the image. Finally, we verify whether the correct user has been traced. When $N > N'$, it can be assumed that some users have been assigned a watermark but have not generated any images.

See Fig. 5b, when $N = 10^6$, Gaussian Shading exhibits almost perfect traceability in seven cases. Although the traceability accuracy for Brightness is only 95.47%, if a user generates two images, the probability of successfully tracing him is still no less than 99%.

4.3. Comparison to Baselines

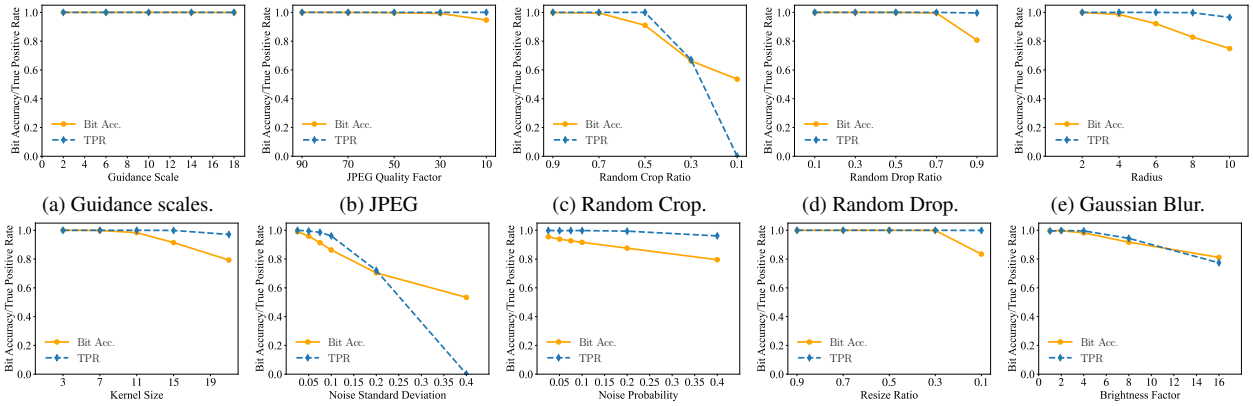
In this section, we compare the performance of Gaussian Shading with baselines on SD V1.4, V2.0, and V2.1. We use our implementations for each method, see details in Supplementary Material.

We conduct tests on $1,000$ generated images for each method respectively. See Tab. 1. Gaussian Shading exhibits strong robustness and significantly outperforms baselines in both scenarios. In terms of bit accuracy, it surpasses the best-performing baseline by approximately 7%. This can be attributed to the extensive diffusion of the watermark throughout the entire latent space, establishing a profound binding between the watermark and the image semantics.

To measure the performance bias introduced by the watermark embedding, we apply a t-test to evaluate. The hypotheses are $H_0 : \mu_s = \mu_0, H_1 : \mu_s \neq \mu_0$, where μ_s and μ_0 represent the average FID [14] or CLIP-Score [29] of multiple sets of watermarked and watermark-free images, respectively. A lower t-value indicates a higher probability that H_0 holds. If the t-value is larger than a threshold, H_0 is rejected, and model performance is considered to have been affected. See Tab. 1, Gaussian Shading achieves the smallest t-value, which indirectly reflects its performance-lossless characteristic. For a detailed analysis of the t-test, please refer to the Supplementary Material.

Methods	Metrics					
	TPR (Clean)	TPR (Adversarial)	Bit Acc. (Clean)	Bit Acc. (Adversarial)	FID (t-value ↓)	CLIP-Score (t-value ↓)
Stable Diffusion	-	-	-	-	25.23±.18	0.3629±.0006
DwtDct [6]	0.825/0.881/0.866	0.172/0.178/0.173	0.8030/0.8059/0.8023	0.5696/0.5671/0.5622	24.97±.19 (3.026)	0.3617±.0007 (3.045)
DwtDctSvd [6]	**1.000/1.000/1.000**	0.597/0.594/0.599	0.9997/0.9987/0.9987	0.6920/0.6868/0.6905	24.45±.22 (8.253)	0.3609±.0009 (4.452)
RivaGAN [45]	0.920/0.945/0.963	0.697/0.697/0.706	0.9762/0.9877/0.9921	0.8986/0.9124/0.9019	24.24±.16 (12.29)	0.3611±.0009 (4.259)
Tree-Ring [41]	**1.000/1.000/1.000**	0.894/0.898/0.906	-	-	25.43±.13 (2.581)	0.3632±.0006 (0.8278)
Stable Signature [10]	**1.000/1.000/1.000**	0.502/0.505/0.496	0.9987/0.9978/0.9979	0.7520/0.7472/0.7500	25.45±.18 (2.477)	0.3622±.0027 (0.7066)
Ours	**1.000/1.000/1.000**	**0.997/0.998/0.996**	**0.9999/0.9999/0.9999**	**0.9753/0.9749/0.9724**	25.20±.22 (**0.3567**)	0.3631±.0005 (**0.6870**)

Table 1. Comparison results. We control the FPR at 10^{-6}, and evaluate the TPR and bit accuracy for SD V1.4/V2.0/V2.1. To assess the bias in model performance, we conduct a t-test on SD V2.1. Adversarial here refers to the average performance of a series of noises. Additional results can be found in Supplementary Material.



(a) Guidance scales. (b) JPEG (c) Random Crop. (d) Random Drop. (e) Gaussian Blur.

(f) Median Filter. (g) Gaussian Noise. (h) Salt and Pepper Noise. (i) Resize. (j) Brightness.

Figure 6. Ablation studies.

4.4. Ablation Studies

In this section, we conduct comprehensive ablation experiments on SD V2.1 to demonstrate hyperparameter selection. Unless specified, we generate $1,000$ images and test the TPR and the bit accuracy with a theoretical FPR of 10^{-6}.

Watermark capacity. The watermark capacity is determined by three parameters: channel diffusion factor f_c, height-width diffusion factor f_{hw}, and embedding rate l. See Tab. 2, to balance the capacity and robustness of Gaussian Shading, we chose $f_c = 1$ and $f_{hw} = 8$. After fixing f_c and f_{hw}, we vary l to examine if it could enhance the capacity, and additional results can be found in Supplementary Material. Considering all factors, we determine that the optimal solution is $f_c = 1$, $f_{hw} = 8$, and $l = 1$, resulting in a watermark capacity of 256 bits.

Sampling methods. To validate the generalization, we select five commonly used sampling methods, all continuous-time samplers based on ODE solvers [32]. See Tab. 3, all of them exhibit excellent performance with a bit accuracy of approximately 97% against noises.

Impact of the inversion step. In practice, the inference step is often unknown, which introduces a mismatch with the inversion step. See Tab. 4, such mismatch introduces minimal loss in accuracy. Considering the high efficiency of existing samplers, the inference step generally does not exceed 50. Therefore, we set the inversion step to 50.

Guidance scales. Given diverse user preferences for image-prompt alignment, larger guidance scales ensure faithful adherence to prompts, while smaller scales grant the model greater creative freedom. In SD, the guidance scale is typically selected from the range of $[5, 15]$. Hence, experiments cover the range of 2 to 18. For the inversion, an empty prompt is used for guidance, and the guidance scale is fixed at 1, assuming unknown information during extraction. In Fig. 6a, the bit accuracy of Gaussian Shading surpasses 99.9%, showing its reliability in real-world-like scenes.

Noise intensities. To further test the robustness, we conduct experiments using different intensities of noises. See Figs. 6b to 6j, for Random Crop and Gaussian Noise, performance declines significantly with higher intensities. However, for the other seven types of noise, even at high intensities, the bit accuracy remains approximately 80%.

4.5. Attacks against Gaussian Shading

We consider two malicious attacks: compression attack, where the attacker employs a neural network to compress watermarked images, and inversion attack, assuming the attacker is aware of the watermark embedding method, enabling them to modify the image's latent representations.

Compression attack. We utilize popular auto-encoders [2, 5, 9, 30] to compare Stable Signature (SS) with Gaussian Shading across various compression rates. Additionally, we assess the compression quality through the PSNR between the compressed and watermarked images. See Figs. 7a

Noise	f_c-f_{hw} (k bits)							
	1-2 (4096)	4-1 (4096)	1-4 (1024)	4-2 (1024)	1-8 (256)	4-4 (256)	1-16 (64)	4-8 (64)
None	0.9413	0.9380	0.9985	0.9980	0.9999	0.9999	1.0000	1.0000
Adversarial	0.7302	0.7238	0.8769	0.8614	0.9724	0.9671	0.9959	0.9953

Table 2. Bit accuracy with different factors f_c and f_{hw} , where $l = 1$. Additional results can be found in Supplementary Material.

Noise	Sampling Methods				
	DDIM [32]	UniPC [48]	PNDM [23]	DEIS [46]	DPMSolver [25]
None	0.9999	1.0000	1.0000	0.9999	0.9999
Adversarial	0.9706	0.9628	0.9721	0.9715	0.9724

Table 3. Bit accuracy with different sampling methods. Additional results can be found in Supplementary Material. These methods differ only in accuracy and order. DDIM is a first-order estimate of the ODE. Accordingly, DDIM inversion ensures a lower bound on the accuracy of the inversion process. Therefore, it can naturally be applied to higher-order and higher-accuracy methods.

Inference Step	Inversion Step			
	10	25	50	100
10	0.9999	0.9999	0.9999	0.9999
25	0.9998	0.9999	1.0000	1.0000
50	0.9995	0.9997	0.9999	0.9999
100	0.9994	0.9996	0.9999	0.9999

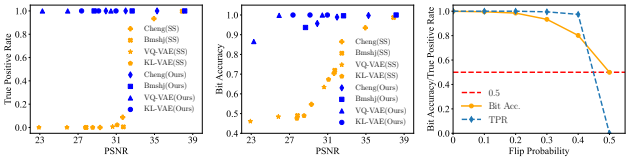
Table 4. Bit accuracy with different inference and inversion step.

and 7b, Gaussian Shading significantly outperforms Stable Signature. This is because Gaussian Shading diffuses the watermark across the entire semantic space of images, while Stable Signature relies solely on the image texture.

Inversion attack. Assuming the attacker is aware of the embedding method, a more effective approach to erasing is through inversion to obtain latent representations and subsequently modify them. We validate the robustness against such attacks. Importantly, our experiments assume the strongest attacker capability of using the same model as Alice for precise inversion. In real-world scenarios, where the watermark embedding is not publicly available, the attacker's capabilities would be weaker.

Specifically, we perform inversion to obtain latent representations and randomly flip a certain rate of them. Using the flipped latent representations, we regenerate the images and extract the watermark. See Fig. 7c. the watermark can still be reliably extracted when the flipping rate (FR) is less than 0.4. At high FRs, significant changes in images are observed. Although the watermark cannot be accurately extracted, we consider the image transformed into a different one, resulting in the content not intended to be protected.

From another perspective, the attacker can launch a forgery attack by performing inversion on an innocuous image from Bob and generating harmful content using a different prompt. See Fig. 7c, when the FR is 0, Alice can accurately trace Bob based on the forgeries, enabling the attacker to successfully frame Bob. Therefore, protecting the model from leakage is crucial for operators.



(a) Detection results. (b) Traceability results. (c) Inversion attack.

Figure 7. Performance of Gaussian Shading under Malicious Attack, where (a) and (b) are under compression attack and (c) is under inversion attack.

5. Limitations

Despite extensive experimental validation of Gaussian Shading's superior performance, our work still has certain limitations. Firstly, the usage scenarios are restricted due to the reliance on DDIM inversion [32], which necessitates the utilization of continuous-time samplers based on ODE solvers [32] like DPMSolver [25]. Secondly, Gaussian Shading employs stream ciphers, necessitating proper key usage and management on the deployment platform. Additionally, we assume that the model is not publicly accessible, and only operators can verify the watermark, providing a certain level of protection against white-box attacks and ensuring security. However, if a legitimate third party requires watermark verification, cooperation from the operators becomes necessary. Lastly, Gaussian Shading is vulnerable to forgery attacks, emphasizing the importance for operators to safeguard the model parameters.

6. Conclusion and Future Work

We propose Gaussian Shading, a provably performance-lossless watermarking applied to diffusion models. Compared to baseline methods, Gaussian Shading offers simplicity and effectiveness by making a simple modification in the sampling process of the initial latent representation. Extensive experiments validate the superior performance in both detection and traceability scenarios. To our knowledge, we are the first to propose and implement a performance-lossless approach in image watermarking.

Regarding future work, we will introduce more efficient inversion methods [40, 44] and include a wider range of sampling methods. Additionally, careful consideration should be given to counteracting forgery attacks.

Acknowledgement. This work was supported in part by the Natural Science Foundation of China under Grant U2336206, 62102386, 62072421, 62372423, and 62121002.

References

[1] Ali Al-Haj. Combined dwt-dct digital image watermarking. *Journal of computer science*, 3(9):740–746, 2007. 3

[2] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018. 7

[3] Daniel J Bernstein et al. Chacha, a variant of salsa20. In *Workshop record of SASC*, pages 3–5. Citeseer, 2008. 4, 5

[4] Kejiang Chen, Hang Zhou, Hanqing Zhao, Dongdong Chen, Weiming Zhang, and Nenghai Yu. Distribution-preserving steganography based on text-to-speech generative models. *IEEE Transactions on Dependable and Secure Computing*, 19(5):3343–3356, 2021. 4

[5] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7939–7948, 2020. 7

[6] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital watermarking and steganography*. Morgan kaufmann, 2007. 2, 3, 6, 7

[7] Yingqian Cui, Jie Ren, Han Xu, Pengfei He, Hui Liu, Lichao Sun, and Jiliang Tang. Diffusionshield: A watermark for copyright protection against generative diffusion models. *arXiv preprint arXiv:2306.04642*, 2023. 2, 3

[8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2

[9] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 7

[10] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. *arXiv preprint arXiv:2303.15435*, 2023. 2, 3, 6, 7

[11] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022. 2

[12] Huiping Guo and Nicolas D Georganas. Digital image watermarking for joint ownership. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 362–371, 2002. 3

[13] Mohamed Hamidi, Mohamed El Haziti, Hocine Cherifi, and Mohammed El Hassouni. Hybrid blind robust image watermarking technique based on dft-dct and arnold transform. *Multimedia Tools and Applications*, 77:27181–27214, 2018. 3

[14] 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. 5, 6

[15] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2

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

[17] Nicholas J Hopper, John Langford, and Luis Von Ahn. Provably secure steganography. In *Advances in Cryptology—CRYPTO 2002: 22nd Annual International Cryptology Conference Santa Barbara, California, USA, August 18–22, 2002 Proceedings 22*, pages 77–92. Springer, 2002. 4, 5

[18] Zhaoyang Jia, Han Fang, and Weiming Zhang. Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression. In *Proceedings of the 29th ACM international conference on multimedia*, pages 41–49, 2021. 3

[19] Varsha Kishore, Xiangyu Chen, Yan Wang, Boyi Li, and Kilian Q Weinberger. Fixed neural network steganography: Train the images, not the network. In *International Conference on Learning Representations*, 2021. 3

[20] Deepa Kundur and Dimitrios Hatzinakos. A robust digital image watermarking method using wavelet-based fusion. In *Proceedings of International Conference on Image Processing*, pages 544–547. IEEE, 1997. 3

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 3

[22] Sunil Lee, Chang D Yoo, and Ton Kalker. Reversible image watermarking based on integer-to-integer wavelet transform. *IEEE Transactions on information forensics and security*, 2 (3):321–330, 2007. 3

[23] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022. 4, 8

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

[25] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 2, 4, 5, 8

[26] Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. Distortion agnostic deep watermarking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13548–13557, 2020. 3

[27] 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. 1

[28] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 2

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2, 5, 7

[31] 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

[32] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020. 2, 4, 5, 7, 8

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

[34] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020. 1

[35] Srdjan Stankovic, Irena Orovic, and Nikola Zaric. An application of multidimensional time-frequency analysis as a base for the unified watermarking approach. *IEEE Transactions on Image Processing*, 19(3):736–745, 2009. 3

[36] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2117–2126, 2020. 3

[37] Min-Jen Tsai, Kuang-Yao Yu, and Yi-Zhang Chen. Joint wavelet and spatial transformation for digital watermarking. *IEEE Transactions on Consumer Electronics*, 46(1):237, 2000. 3

[38] Ron G Van Schyndel, Andrew Z Tirkel, and Charles F Osborne. A digital watermark. In *Proceedings of 1st international conference on image processing*, pages 86–90. IEEE, 1994. 2, 3

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[40] 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. 8

[41] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *arXiv preprint arXiv:2305.20030*, 2023. 2, 3, 6, 7

[42] Cheng Xiong, Chuan Qin, Guorui Feng, and Xinpeng Zhang. Flexible and secure watermarking for latent diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1668–1676, 2023. 2, 3

[43] Chaoning Zhang, Philipp Benz, Adil Karjauv, Geng Sun, and In So Kweon. Udh: Universal deep hiding for steganography, watermarking, and light field messaging. *Advances in Neural Information Processing Systems*, 33:10223–10234, 2020. 3

[44] Jiaxin Zhang, Kamalika Das, and Sricharan Kumar. On the robustness of diffusion inversion in image manipulation. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*, 2023. 8

[45] Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust invisible video watermarking with attention. *arXiv preprint arXiv:1909.01285*, 2019. 2, 3, 6, 7

[46] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022. 4, 8

[47] Weiming Zhang, Kejiang Chen, and Nenghai Yu. Provable secure steganography: Theory, application and prospects. *Journal of Cybersecurity*, 1:38–46, 2023. 4

[48] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *arXiv preprint arXiv:2302.04867*, 2023. 4, 8

[49] 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. 2, 3

[50] Xin Zhong, Pei-Chi Huang, Spyridon Mastorakis, and Frank Y Shih. An automated and robust image watermarking scheme based on deep neural networks. *IEEE Transactions on Multimedia*, 23:1951–1961, 2020. 3

[51] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 657–672, 2018. 3