Revised manuscript (with changes marked)

TraceMark-LDM: Authenticatable Watermarking for Latent Diffusion Models via Binary-Guided Rearrangement

Wenhao Luo[a], Zhangyi Shen[a,*], Ye Yao[a], Feng Ding[b], Guopu Zhu[c] and Weizhi Meng[d]

[a]School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China
[b]Nanchang University, Nanchang, JX 330031, China
[c]School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
[d]School of Computing and Communications, Lancaster University, United Kingdom

ARTICLE INFO

Keywords:
Image attribution
Trustworthy AIGC
Diffusion models

ABSTRACT

Image generation algorithms are increasingly integral to diverse aspects of human society, driven by their practical applications. However, insufficient oversight in artificial intelligence-generated content (AIGC) can facilitate the spread of malicious content and increase the risk of unauthorized use. Among the diverse range of image generation models, the Latent Diffusion Model (LDM) is currently the most widely used, dominating the majority of the Text-to-Image model market. Currently, most attribution methods for LDMs rely on directly embedding watermarks into the generated images or their intermediate noise, a practice that compromises both the quality and the robustness of the generated content. To address these limitations, we introduce TraceMark-LDM, a novel algorithm that integrates watermarking to attribute generated images while guaranteeing non-destructive performance. Unlike current methods, TraceMark-LDM leverages watermarks as guidance to rearrange random variables sampled from a Gaussian distribution. To mitigate potential deviations caused by inversion errors, the small-magnitude elements are grouped and strategically rearranged. Additionally, we fine-tune the LDM encoder to enhance the robustness of the watermark. Experimental results show that images synthesized using TraceMark-LDM exhibit superior quality and attribution accuracy compared to state-of-the-art (SOTA) techniques. Notably, TraceMark-LDM demonstrates exceptional robustness against various common attack methods, consistently outperforming SOTA methods. Our code is available at https://github.com/luowhDevSpace/TraceMark.

1. Introduction

Text-to-Image generation enables creating images based on descriptive text prompts (Xu et al., 2018; Qiao et al., 2019; Song and Ermon, 2019; Song et al., 2021b). This technology has significantly impacted various fields, such as art, design, and marketing (Frolov et al., 2021). Currently, the Latent Diffusion Model (LDM) (Rombach et al., 2022; Fan et al., 2024) is particularly notable among generative models due to its efficiency in generating high-quality images. However, the widespread accessibility of this technology (Dhariwal and Nichol, 2021; Nichol and Dhariwal, 2021; Gu et al., 2022; Ho and Salimans, 2022) allows the general public to generate large volumes of synthetic data, which may compromise the reliability of images as indispensable carriers of information (Guo et al., 2023). To mitigate this concern, it is essential to implement methods for the attribution of synthesized images.

Watermarking has recently emerged as a promising approach for achieving this objective (Fernandez et al., 2023; Wen et al., 2023; Liu et al., 2024; Yuan et al., 2024; Yang et al., 2024; Li et al., 2024; He et al., 2020; Qin et al., 2024; Zhong et al., 2020; Fang et al., 2022). In the literature, numerous watermarking-based algorithms have been developed to authenticate images synthesized by LDMs. These algorithms incorporate the watermark into the generated image in various ways, with a specific watermark decoder used to verify the embedded watermark as well as determine the image's attribution. Broadly, these methods can be categorized into three types: post-generation watermarking, in-generation watermarking, and initial noise sampling watermarking.

Post-generation watermarking (Cox et al., 2007; Zhang et al., 2019; Yin et al., 2022; Bui et al., 2022) embeds watermarks directly into synthesized images. However, when the embedded payload becomes large (e.g., containing detailed trademark data and user identity information), increased payload size can introduce visible artifacts and degrade image quality. In-generation watermarking embeds watermarks during denoising in the latent space or decoding into the pixel space. For example, AquaLoRA (Feng et al., 2024) embeds watermarks during denoising by fine-tuning the U-Net structure. In addition, some researchers (Fernandez et al., 2023; Cui et al., 2025; Xiong et al., 2023; Meng et al., 2025; Rezaei et al., 2024) have processed denoising latent variables by watermark embedding module or fine-tuning a pre-trained variant autoencoder (VAE) (Kingma and Welling, 2014) decoder to embed watermarks. However, this approach has a higher computational cost.

Both post-generation and in-generation watermarking methods may compromise output quality, as they disrupt the proper distribution of the desired output. Furthermore, they are fragile to diffusion-based regeneration attacks (Zhao et al., 2024). Embedding the watermark into the initial noise

*Corresponding author.
luowenhao@hdu.edu.cn (Wenhao Luo); shenzhangyi@hdu.edu.cn (Zhangyi Shen); yaoye@hdu.edu.cn (Ye Yao); fengding@ncu.edu.cn (Feng Ding); guopu.zhu@hit.edu.cn (Guopu Zhu); weizhi.meng@ieee.org (Weizhi Meng)

overcomes these limitations because direct interference of the output is avoided. Also, this algorithm maintains compatibility with any diffusion model without retraining. However, current implementations exhibit drawbacks: Tree-Ring (Wen et al., 2023) embeds a predefined key into the Fourier transform of the initial latent noise, resulting in a deviation from a Gaussian distribution and consequently impairing model performance. Gaussian Shading (Yang et al., 2024) converts watermark information into the initial latent noise through distribution-preserving sampling. However, the official implementation's ChaCha20 step is time-consuming, modestly reducing generation efficiency. Similarly, recent schemes such as PRCWM (Gunn et al., 2025) and T2SMark (Yang et al., 2025) fundamentally rely on precise inversion-based recovery of designated test bits or keys. Under common distortion conditions, increased DDIM inversion errors lead to failed test bit/key recovery, ultimately causing the message-level extraction breakdown.

To address limitations of existing LDM watermarking methods, we propose TraceMark-LDM, a non-destructive watermarking scheme. The proposed method encodes watermark bits via a binary-guided rearrangement of Gaussian samples to construct the watermarked initial noise. The generative backbone remains unchanged. To maintain positional randomness, we use a seed-controlled composition of permutations that restores random element placement. The design avoids the distribution shift reported for Tree-Ring and dispenses with the time-consuming procedure in Gaussian Shading. To counter the increased inversion error caused by image distortions, we fine-tune the LDM encoder to align corrupted and clean latents, thereby improving robustness under diverse perturbations and diffusion-based regeneration attacks. Fig. 1 illustrates the application scenarios of TraceMark-LDM. In summary, the contributions of this paper are as follows:

1. We propose a general watermarking framework for latent diffusion models (LDMs) that embeds watermarks by rearranging Gaussian noise samples. This approach preserves the LDM's high-quality image generation while reducing watermark extraction errors.
2. We fine-tune the LDM encoder to better align encoded latent variables with their original counterparts. This adjustment reduces inversion errors and substantially improves the robustness of watermark extraction under various attack scenarios.
3. We conduct extensive experiments across multiple attack types and intensity levels to demonstrate TraceMark-LDM's effectiveness and robustness; ablation studies further confirm the efficacy of encoder fine-tuning.

The remainder of this paper is organized as follows. Section 2 reviews related works. Section 3 discusses the motivation behind our method and outlines the proposed approach. Section 4 details the experimental setup and results. Finally, the conclusion is given in Section 5.

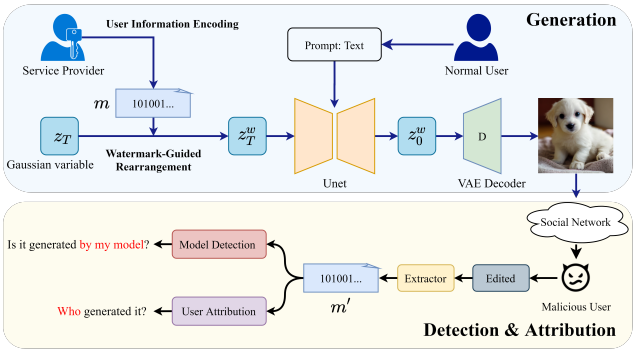


Fig. 1: Application scenarios for TraceMark-LDM. The service provider embeds a digital watermark, encoded with user-specific information, into the generated image. Depending on requirements, the watermark can facilitate either model detection (verifying whether an image originates from the service model) or user attribution (identifying the responsible user). In cases of malicious tampering or unauthorized use of protected works, the extracted watermark enables reliable model verification and user attribution.

2. Related Works

2.1. Diffusion Models

The prototype of diffusion models was introduced by Sohl-Dickstein et al. (2015), who proposed a deep unsupervised learning algorithm grounded in non-equilibrium statistical physics. The model generates samples from a target distribution through a forward and backward Markov chain, but suffers from high computational cost and limited sample quality. Ho et al. (2020) advanced this line of work by proposing the Denoising Diffusion Probabilistic Model (DDPM), which improves training stability and sample diversity via variational inference and a carefully designed noise schedule. However, DDPM remains computationally expensive due to its slow sampling process. To address this, Song et al. (2021a) introduced the Deterministic Denoising Diffusion Implicit Model (DDIM), which reformulates the reverse process to allow step skipping, enabling faster generation of high-quality images. Building upon these advancements, Rombach et al. (2022) proposed the Latent Diffusion Model (LDM), which performs the diffusion process in a lower-dimensional latent space using a pre-trained autoencoder. This significantly reduces computational cost while enabling high-resolution image synthesis, leading to notable improvements in both efficiency and sample quality.

2.2. Deep Learning Watermarking

As a pioneering deep learning-based watermarking framework, HiDDeN (Zhu et al., 2018) achieves imperceptible watermark embedding through joint training of encoder and decoder networks, demonstrating strong robustness against distortions such as blurring, cropping, and compression. RivaGAN (Zhang et al., 2019) adopts a GAN-based architecture enhanced with attention mechanisms and adversarial training to improve resistance to distortion and compression artifacts. To enhance robustness against JPEG

compression, MBRS (Jia et al., 2021) proposes an end-to-end training framework that incorporates randomized real JPEG compression, simulated JPEG artifacts, and noise-free layers as adversarial components. This framework further integrates Squeeze-and-Excitation (SE) modules for improved feature extraction and dedicated message processors for efficient information embedding. Meanwhile, CIN (Ma et al., 2022) combines reversible and non-reversible modules: the former ensures high concealment capability, while the latter enhances robustness under strong noise attacks. Zhang et al. (2024) introduced EditGuard, a proactive framework that integrates copyright protection with tamper-agnostic localization.

2.3. Watermarking in Generative Models

Detecting and tracing content generated by generative models can be achieved by covertly embedding digital watermarks into the generated outputs. For instance, Stable Diffusion adopts traditional digital image watermarking techniques (Van Schyndel et al., 1994; Badran et al., 2009; Mohammed et al., 2014; Qin et al., 2017; Fan et al., 2021; Luo et al., 2024; Kang et al., 2003; Cox et al., 2007; Zhang et al., 2019; Fang et al., 2019; Wang et al., 2022; Huan et al., 2021; Luo et al., 2026), such as DwtDct and DwtDctSvd, which can be directly applied to synthesized images. In addition, deep learning-based methods have also been explored for watermarking generated content. Recently, researchers (Xiong et al., 2023) have developed watermarking techniques tailored for diffusion models to improve robustness. Fernandez et al. (2023) proposed fine-tuning the LDM decoder and employing a pre-trained extractor to retrieve watermarks from generated images. However, these methods embed watermarks by modifying the host image. When the perturbations are weak, robustness against distortion is limited; whereas stronger perturbations enhance robustness but degrade image quality.

To address the common conflict between image quality and watermark robustness, Meng et al. (2025) designed Latent Watermark, a unified framework for watermark embedding and detection within the latent space, and proposed a progressive training strategy. This approach weakens the direct conflict between robustness and generation quality, thereby alleviating their contradiction. However, the image quality still faces challenges when more bits of watermarks are embedded. Wen et al. (2023) introduced an approach to enhance resistance to corruptions and perturbations by embedding a specially designed pattern in the Fourier transform domain of the initial latent noise. Ci et al. (2024) extended this approach to support multiple keys. However, these methods corrupt the Gaussian distribution of the initial noise, limiting the randomness of the sampling process at the expense of the performance of the original generated model. To overcome these limitations, Yang et al. (2024) converts watermark information into the initial noise for the generative model through steps such as watermark diffusion, watermark randomization, and distribution-preserving sampling. This approach embeds watermarks without impairing image quality. Nevertheless, the use of Chacha20 encryption during the watermark randomization process is time-consuming, potentially resulting in a decrease in the speed of generation. PRCWM (Gunn et al., 2025) constructs the initial noise via a pseudorandom error-correcting code, aiming for cryptographic indistinguishability and image-quality preservation. T2SMark (Yang et al., 2025) adopts two-stage tail-truncated sampling that embeds bits in high-confidence tails, samples the central region to preserve diversity, and integrates a session key in a dedicated channel. Both methods' detection depends on accurately recovering specified test bits or keys from inversion noise. Under common distortions, larger inversion errors readily cause recovery deviations of test bits or session keys, leading to failed message extraction.

3. Proposed Method

In this section, we propose a general watermarking framework for latent diffusion models (LDMs), called TraceMark-LDM, which embeds watermarks by rearranging Gaussian noise samples. First, we analyze the impact of DDIM inversion on the noise and how element-wise shuffling affects the overall distribution in section 3.1. Subsequently, we present the watermark embedding process in section 3.2. Section 3.3 provides a detailed description of the fine-tuning procedure for the LDM encoder. Finally, in section 3.4, we explain the watermark extraction process. The notations that will be used in this paper are listed in Table 1.

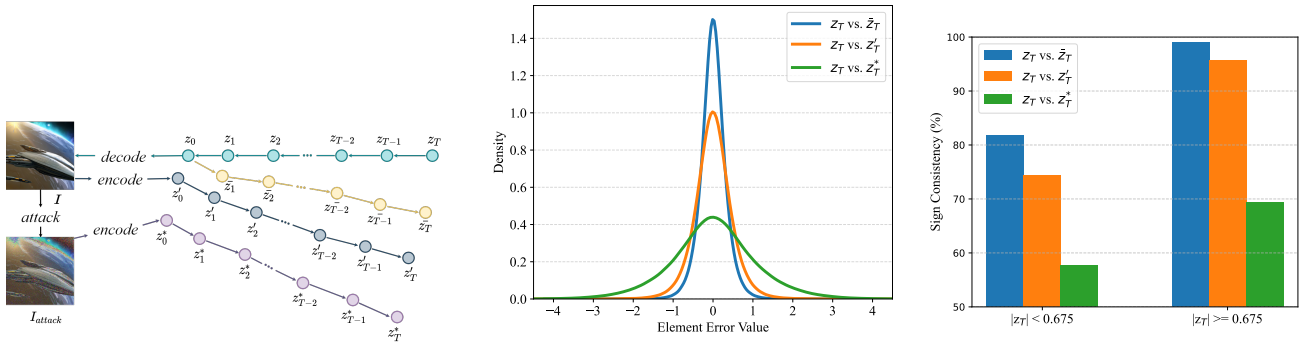
3.1. Preliminary Analysis

To develop a robust, lossless watermark embedding strategy, we systematically analyze the denoising and inversion processes inherent in diffusion models, which yield the following key observations:

1. **Minimizing the difference in latent representation of distorted images helps reduce inversion errors.** As illustrated in Fig. 2a, decoding the denoised latent variable $\mathbf{z}_0$ into an image I and then re-encoding it via the VAE encoder yields a recovered latent representation $\mathbf{z}'_0$, where $\mathbf{z}'_0 \neq \mathbf{z}_0$, indicating that the encoder–decoder introduces distortion in the latent space. Recent research on numerical errors in diffusion ODE inversion (Hu et al., 2024) has pointed out that the bidirectional mapping process between $\mathbf{z}_0$ and intermediate latents $\mathbf{z}_T$ is highly fragile, and even minor discrepancies can accumulate during the inversion process. This suggests that encoding–decoding operations increase inversion errors. Moreover, when the image is subjected to an adversarial attack, the perturbed image I_{attack} leads to a latent representation $\mathbf{z}^*_0$, obtained by encoding I_{attack}, which deviates further from the clean latent $\mathbf{z}_0$, exacerbating the inversion error.

In Fig. 2b, we quantify these inversion errors under various conditions. Three representative trajectories are considered:

(1) Starting from the original latent $\mathbf{z}_0$, DDIM inversion is applied to obtain the re-noised latent $\bar{\mathbf{z}}_T$.



(a) (b) (c)

Fig. 2: Illustration of the denoising and inversion process and analysis of inversion errors. (a) Visualizes the generation process from noise $\mathbf{z}_T$ to latent variable $\mathbf{z}_0$ and decoding to an image, alongside three inversion trajectories starting from the original latent $\mathbf{z}_0$, the decode–re–encoded latent z'_0, and the perturbed image's latent $\mathbf{z}^*_0$. (b) Presents the distribution of inversion errors under different conditions, showing the error values between $\mathbf{z}_T$ and its reconstructed counterparts ($\bar{\mathbf{z}}_T$, $\mathbf{z}'_T$, and $\mathbf{z}^*_T$). (c) Evaluates the sign consistency of latent elements, highlighting that large-magnitude elements ($|\mathbf{z}_T| \geq 0.675$) exhibit higher robustness to inversion errors compared to small-magnitude ones ($|\mathbf{z}_T| < 0.675$).

(2) The latent $\mathbf{z}'_0$ is obtained by decoding $\mathbf{z}_0$ into an image I and then re-encoding it via the VAE encoder; DDIM inversion is subsequently performed to produce $\mathbf{z}'_T$.

(3) A perturbed image I_{attack} is encoded to yield $\mathbf{z}^*_0$, from which DDIM inversion is applied to generate $\mathbf{z}^*_T$.

Notably, when bypassing VAE encoding–decoding, recovery from $\mathbf{z}_0$ to $\bar{\mathbf{z}}_T$ exhibits the smallest symmetric deviation, with errors primarily confined to the range $[-0.6, 0.6]$ around zero. These findings indicate that aligning the perturbed latent $\mathbf{z}^*_0$ with the clean reference $\mathbf{z}_0$ effectively mitigates both distortion and inversion-induced errors.

2. **The signs after denoising and inversion of large-magnitude elements have high stability.** Based on our analysis of inversion errors, we categorized each element of the initial noise by the quartiles of the standard normal distribution, grouping them into large-magnitude elements ($|\mathbf{z}_T| \geq 0.675$) and small-magnitude elements ($|\mathbf{z}_T| < 0.675$). We then statistically evaluated the consistency of their signs after denoising and inversion. As shown in Fig. 2c, large-magnitude elements retain over 95% sign consistency following inversion, and even under distortion, this consistency remains close to 70%. This robustness makes them well-suited for encoding watermark information. In contrast, small-magnitude elements exhibit significantly lower sign consistency, rendering them unsuitable for watermark encoding on an individual basis.

3. **Distributional invariance of i.i.d. Gaussian samples under random permutations.** Let $X = (X_1, \ldots, X_n)$ be a random vector with independent and identically distributed (i.i.d.) standard Gaussian components. Let ϕ denote the standard normal density. Hence, the joint probability density function (pdf) is

$$f_X(\mathbf{x}) = \prod_{i=1}^{n} \phi(x_i) \tag{1}$$

Let $\mathcal{P}$ be a permutation drawn uniformly at random from the symmetric group S_n (the set of all permutations on n elements), independent of X, and define the permuted vector. $X' = \Pi_{\mathcal{P}} X$. By exchangeability of i.i.d. Gaussian components, for any fixed $\sigma \in S_n$,

$$(X_{\sigma(1)}, \ldots, X_{\sigma(n)}) \overset{d}{=} (X_1, \ldots, X_n) \tag{2}$$

where $\overset{d}{=}$ denotes equality in distribution. Marginalizing over $\mathcal{P}$ yields

$$f_{X'}(\mathbf{x}) = \sum_{\sigma \in S_n} f_{\Pi_\sigma X}(\mathbf{x}) \frac{1}{n!} = \sum_{\sigma \in S_n} f_X(\mathbf{x}) \frac{1}{n!} \quad = f_X(\mathbf{x}) \tag{3}$$

In conclusion, after applying a uniformly random, independent permutation, the resulting vector X' retains the original distribution of X, i.e., $X' \overset{d}{=} X$.

Based on the sign consistency and distributional invariance discussed above, we propose a novel watermarking algorithm for latent diffusion models (LDM), as illustrated in Fig. 3. The framework consists of three key phases: watermark embedding, LDM encoder fine-tuning, and watermark extraction. In the embedding phase, a random variable sampled from a Gaussian distribution is partitioned into four subsets according to its element values. These subsets are then rearranged based on the watermark bits and fed into the generative model to synthesize content. The encoder fine-tuning phase aims to mitigate errors in the latent representation caused by potential attacks or perturbations. During the extraction phase, the generated image is inverted back to its initial latent noise via DDIM inversion, enabling watermark retrieval based on the predefined arrangement. The following sections detail the implementation of the proposed approach.

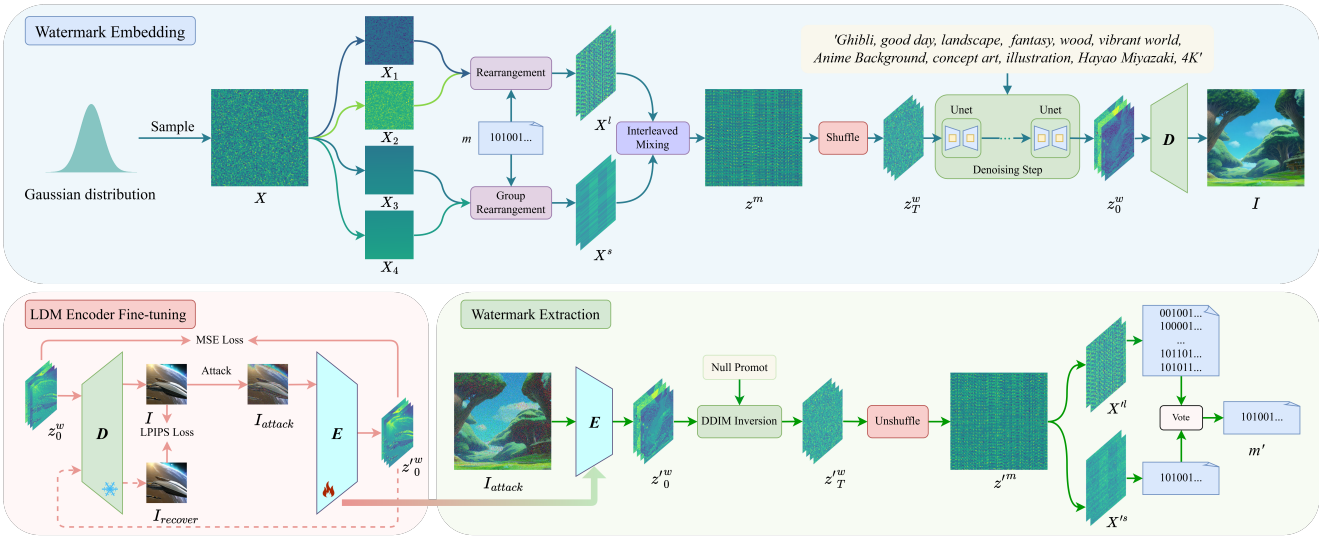


Fig. 3: The framework of TraceMark-LDM. The watermark information m is used to guide a rearrangement process to produce a latent input $\mathbf{z}_T^w$ for generating watermarked image I. During the fine-tuning phase, the decoder parameters are frozen and only the encoder parameters are updated. The extraction process is an inversion of the embedding.

Table 1
Summary of notations in this paper.

Notation	Description
X	The randomly sampled Gaussian samples
m	The embedded watermark
k	The length of the embedded watermark
s	The secret key used for shuffling
N, P	The positive and negative subsets of X
$X_{1,2,3,4}$	The subsets divided by the quartiles of X
R	The small-magnitude elements in X
G_n, G_p	The subsets of positive and negative groups composed of R
X^l	The rearranged X_1 and X_2 sequence
X^s	The group rearranged X_3 and X_4 sequence
$\mathbf{z}_T^w$	The watermarked latent initial noise
$\mathbf{z}_0^w$	The denoised latent variable
I	The watermarked image
I_{attack}	The attacked watermarked image
E	The LDM encoder
D	The LDM decoder

3.2. Watermark Embedding

To maintain the high quality of the generated images, we sample a random variable $X = \{x_1, x_2, x_3, \ldots, x_{(r-1)}, x_r\}$ containing r elements following a Gaussian distribution. Here, $r = c \times h \times w$, where c, h, and w are the dimensions of the latent space in the LDM. The watermark m is a random bitstream of k bits, equally distributed between 0 and 1.

The core idea of our watermark embedding strategy exploits the inherent characteristics of noise in diffusion models, specifically using the sign of noise elements to encode watermark bits. Therefore, the random variables X are divided into a negative set N and a non-negative set P:

$$\begin{aligned} N &= \{x_i \in X \mid x_i < 0\} \\ P &= \{x_i \in X \mid x_i \geq 0\} \end{aligned} \tag{4}$$

3.2.1. Binary Guided Rearrangement

According to the previous analysis, large-magnitude elements are less susceptible to denoising and inversion, and a single element can be used to indicate watermarking information. Based on the division of standard normal distribution quartiles, we select large-magnitude elements from the sets N and P, with each set accounting for a quarter of the total number of elements. These are denoted as X_1 and X_2, respectively.

$$\begin{aligned} X_1 &= \{x_i \in N \mid rank(x_i, N) \leq r/4\} \\ X_2 &= \{x_i \in P \mid rank(x_i, P) \leq r/4\} \end{aligned} \tag{5}$$

Here, $rank(x, S)$ represents the rank of x within the set S when sorted by absolute value in descending order.

The watermark information m directs the rearrangement of elements within the X_1 and X_2 sets. Specifically, if the watermark bit is 0, an element from X_1 is selected; if the watermark bit is 1, an element from X_2 is chosen. To enhance the fault tolerance of the watermark, m is repeatedly used, resulting in a sequence X^l consisting of large-magnitude elements.

3.2.2. Binary Guided Group Rearrangement

As shown in Fig. 2c, under image distortion, only about 55% of the small-magnitude elements preserve their original sign after inversion. This high degree of instability indicates that relying on the sign of individual elements for watermarking substantially weakens robustness. Therefore, it is more effective to exclude such elements during the embedding process.

To fully exploit the potential of these small-magnitude elements, we introduce a group rearrangement strategy. Instead of treating them individually, we logically organize them into groups, ensuring that the watermark information is conveyed through the sign of the sum of each group. This strategy enhances the authentication performance by mitigating the impact of sign flipping at the individual element level. The specific operations are as follows:

The remaining small-magnitude elements in X are combined to form set R with $\frac{r}{2}$ elements, defined as:

$$R = \{x_i \in N \mid x_i \notin X_1\} \cup \{x_i \in P \mid x_i \notin X_2\} \quad (6)$$

Due to stochastic sampling, positive and negative elements in R may be imbalanced. To address this, R is sorted in ascending order and equally split into two subsets, X_3 and X_4, each with $\frac{r}{4}$ elements:

$$\begin{aligned} X_3 &= \{x_i \mid x_i \in sorted(R)[1, r/4]\} \\ X_4 &= \{x_i \mid x_i \in sorted(R)[r/4+1, r/2]\} \end{aligned} \quad (7)$$

Subsequently, X_3 and X_4 are partitioned into subsets G_n and G_p using a symmetric grouping strategy. Taking X_3 as an example, the sorted sequence is divided into $\frac{k}{2}$ groups. Here, k denotes the total length of the watermark. Elements are alternately selected from both ends and dynamically assigned to minimize inter-group sum differences, forming G_n. X_4 is processed in the same manner to obtain G_p.

Compared to using the signs of individual small-magnitude elements for watermark encoding, the aggregated sign of grouped elements offers greater robustness. For example, a group in G_n composed of multiple negative elements yields an initial sum v encoding watermark bit 0. During extraction, although element values may deviate due to inversion errors (as shown in Fig. 2b, where the errors follow a zero-mean symmetric distribution), positive and negative deviations partially cancel during summation. Since v reflects the cumulative contribution of consistently signed elements (e.g., a strongly negative sum for negative groups), its sign is more resistant to noise and inversion-induced perturbations. In other words, group rearrangement transforms many unstable individuals into a stable collective by leveraging the symmetry of the error.

Leveraging this symmetry-driven optimization, watermark embedding proceeds via group-wise rearrangement: if a watermark bit is 0, an unused group from G_n is selected; if 1, a group from G_p is chosen. Selected groups are sequentially concatenated to form the encoded sequence X^s. The resulting sequence X^s is a rearrangement of R that both preserves the distributional properties of the elements and implicitly carries watermarking information through the sequence of signs of the group sum.

3.2.3. *Interleaved Mixing and Shuffle*

The r elements from X^l and X^s are integrated using an interleaved mixing strategy to construct $\mathbf{z}^m$, which serves to preliminarily disrupt the original feature ordering. In this process, elements from X^l and X^s are alternately selected in fixed-length segments to form $\mathbf{z}^m$, following the structure defined by the watermark bits. The resulting sequence is then shuffled and reshaped using the model key s, yielding the initial latent noise $\mathbf{z}_T^w$.

3.2.4. *Image Generation*

After denoising steps, the latent variable $\mathbf{z}_0^w$ for generating the image is obtained. It is subsequently mapped to the spatial domain through the LDM decoder, generating the image $I = D(\mathbf{z}_0^w)$.

3.3. LDM Encoder Fine-Tuning

Initial noise sampling watermarking methods rely on DDIM to invert generated images back to the initial noise. However, inversion errors introduced during this process can significantly degrade watermark robustness. To address this issue, we propose fine-tuning the LDM encoder to better adapt to complex distortion scenarios, thereby reducing inversion errors. Since the encoder solely maps images from pixel space back to latent space and does not participate in the generative process, this fine-tuning does not compromise image quality. As previously observed, directly inverting the original latent variables yields the smallest inversion errors, suggesting that minimizing the discrepancy between the latent variables of distorted and original images can effectively suppress inversion errors and enhance watermark robustness.

Initially, the image $I = D(\mathbf{z}_0^w)$ is generated by decoding the original latent variable $\mathbf{z}_0^w$ using the LDM decoder D. Subsequently, random perturbations are applied to the image I to simulate a distortion scenario, resulting in the perturbed image I_{attack}.

Next, the perturbed image I_{attack} is fed into the fine-tuned LDM encoder E, which maps it back into the latent space to obtain the latent variable $\mathbf{z}_0^* = E(I_{attack})$. The mean squared error (MSE) loss between $\mathbf{z}_0$ and $\mathbf{z}_0^*$ is calculated as follows:

$$\mathcal{L}_{\text{inv}} = \min_{\theta} \mathbf{E}\left[\|\mathbf{z}_0 - E_\theta(I_{attack})\|^2\right] \quad (8)$$

By optimizing the MSE loss, we minimize the difference between the latent variables of the generated image before and after perturbation, thereby reducing the inversion errors of the initial noise.

When using only the MSE inversion loss on $\mathbf{z}_0$, note that $\mathbf{z}_0 \in \mathbb{R}^{c\times h\times w}$ is a high-dimensional tensor. The optimizer often focuses on the largest local errors—i.e., a few channels or spatial locations—while leaving other elements under-optimized. As a result, even if $\|\mathbf{z}_0 - \mathbf{z}_0^*\|^2$ is small overall, the uncorrected channels or pixel regions can drift significantly during the forward DDIM inversion, causing the intermediate latent $\mathbf{z}_T$ to diverge markedly from its original trajectory. To impose consistency across all channels and spatial positions at a global scale, we introduce the perceptual similarity (LPIPS) loss as an additional term.

By feeding $\mathbf{z}_0^*$ into the decoder D, converting it back to the spatial domain to synthesize the recovered image $I_{recover} = D(\mathbf{z}_0^*)$. The LPIPS loss between the original

image I and the recovered image $I_{recover}$ is then calculated as follows:

$$\mathcal{L}_{\text{sim}} = \min_{\theta} \mathbf{E}\left[LPIPS(D(\mathbf{z}_0^w), D(E_\theta(I_{attack})))\right] \quad (9)$$

Finally, the total loss function is optimized to balance two loss terms. This ensures that the fine-tuning minimizes differences between initial noise $\mathbf{z}_0$ and $\mathbf{z}_0^*$ under various perturbations, without introducing additional noise during the inversion process of the diffusion model:

$$\mathcal{L} = \mathcal{L}_{\text{inv}} + \lambda \mathcal{L}_{\text{sim}} \quad (10)$$

where λ is a weight coefficient used to balance the two loss terms.

It is worth noting that the fine-tuning process does not require the diffusion model's inversion steps. This not only significantly improves the efficiency of fine-tuning but also greatly reduces computational costs.

3.4. Watermark Extraction Process

The watermark extraction process is essentially the reverse of the embedding process. We begin by feeding the generated image into the fine-tuned LDM encoder to retrieve the latent variable $\mathbf{z}_0'$, denoted as $\mathbf{z}_0'^w = E'(I_{attack})$. Next, the initial watermarked noise $\mathbf{z}_T'^w$ is recovered through DDIM inversion. The unshuffle operation using the model key s in the embedding process is then applied to $\mathbf{z}_T'^w$ to obtain the interleaved noise $\mathbf{z}'^m$. This conversion aims to reverse the random permutation applied during the embedding process. Subsequently, we perform the deinterleave operation on $\mathbf{z}'^m$, separating it into two parts: X'^l and X'^s.

For each element in X'^l, the watermark bit is determined by its sign: negative elements map to bit 0, and non-negative elements map to bit 1. Because this procedure is repeated on the large-magnitude elements, it produces T copies of the watermark. Similarly, for each group of elements in X'^s, we take the sign of the group sum to obtain one additional copy of the watermark.

In total, $T+1$ copies of the watermark are obtained. A per-bit majority voting is then applied across the copies: for each bit, if more than half of the copies indicate 1, that bit is set to 1; otherwise, it is set to 0. Applying this voting to all bits yields the recovered watermark m'. Majority voting consolidates redundant copies so that isolated per-copy errors are outvoted by the consensus, thereby lowering the bit-error rate and improving decoding robustness.

4. Experiments

In this section, the proposed Tracemark-LDM is evaluated against several state-of-the-art methods. The experimental settings are provided in Section 4.1. The evaluation metrics are defined in Section 4.2. Section 4.3 presents a comparison of our method with other approaches. Finally, the ablation study is conducted in Section 4.4.

Table 2
Attack Method and Parameter Ranges

ID	Attack Method	Parameter Ranges
A	Median Filter	Kernel size: 3 - 19
B	JPEG Compression	Quality factor: 10 - 90
C	Gaussian Blur	Radius: 2.0 - 10.0
D	Gaussian Noise	Standard deviation: 0.05 - 0.25
E	Salt and Pepper Noise	Probability: 0.05 - 0.4
F	Resize and Restore	Scaling factor: 0.1 - 0.9
G	Regen-VAE-A	Quality level: 1 - 5
H	Regen-VAE-B	Quality level: 1 - 5
I	Regen-Diffusion	Noise steps: 300 - 700

4.1. Experimental Settings

4.1.1. Implementation details

We evaluated the efficacy of TraceMark-LDM by employing Stable Diffusion v2.1 as the foundational model. The resolution of the generated watermarked images is set to 512×512, with latent space dimensions of $4 \times 64 \times 64$. Prompts from the Stable-Diffusion-Prompt repository are used in conjunction with the DPMsolver (Lu et al., 2022) multistep scheduler, utilizing a 50-step sampling process and a default guidance scale of 7.5. During the watermark extraction phase, DDIM inversion is performed with the same number of inversion steps, null prompts, and a guidance scale of 1.

To fine-tune the encoder, we simulate distortion environments by applying six attack types (A–F) to 200 clean images, with their definitions and parameters specified in Table 2. The training process uses the AdamW optimizer with a learning rate of 1×10^{-6}, and the loss function weight coefficient λ is set to 1. The model is trained for 100 epochs.

All experiments are performed with PyTorch 2.1.1. The GPU is an NVIDIA RTX 3090 Ti with 24 GB of memory.

4.1.2. Baseline Methods

Several post-generation watermarking techniques, including DwtDct (Cox et al., 2007), DwtDctSvd (Cox et al., 2007), and RivaGAN (Zhang et al., 2019), are compared with our TraceMark-LDM. For in-generation watermarking methods, we employ Stable Signature (SS) (Fernandez et al., 2023), LaWa (Rezaei et al., 2024) and Latent Watermark (LW) (Meng et al., 2025). For initial noise sampling watermarking methods, we adopt Tree-Ring (Wen et al., 2023), RingID (Ci et al., 2024), Gaussian Shading (GS) (Yang et al., 2024), PRCWM (Gunn et al., 2025) and T2SMark (Yang et al., 2025). To ensure a fair comparison, we standardize the watermark length of TraceMark-LDM to 256 bits.

4.2. Evaluation Metrics

4.2.1. Watermark Accuracy

In order to evaluate the performance of watermark detection, we adopt the True Positive Rate at a False Positive Rate of 10^{-6} (TPR@10^{-6} FPR) as the primary metric. TPR@10^{-6} FPR is defined as the detection accuracy

calculated using the corresponding threshold that ensures the theoretical false positive rate remains below 10^{-6}. We assume that the matching or mismatching between the injected watermark bits and the extracted watermark bits are independent and identically distributed variables following a Bernoulli distribution with a probability parameter of 0.5. For comparison methods utilizing watermark lengths of 32, 48, and 256 bits, the respective thresholds are set to 30, 41, and 167 bits. The detection accuracy is computed using the following formula:

$$\text{TPR} = \frac{1}{N}\sum_{n=1}^{N}\mathbb{I}\left(\sum_{k=1}^{K}\left(m_{k,n} \equiv m'_{k,n}\right) > \tau\right) \tag{11}$$

where N is the total number of samples, K is the watermark length, $m_{k,n}$ and $m'_{k,n}$ are the k-th watermark bits injected and extracted for the n-th sample respectively, τ is the threshold, and $\mathbb{I}\{\cdot\}$ is the indicator function, which equals 1 if the condition within the braces is satisfied, and 0 otherwise.

In the context of attribution watermarking, we use the average bit accuracy of all extracted watermark samples as the performance metric to evaluate our method. The specific formula is defined as follows:

$$\text{BA} = \frac{1}{N \cdot K}\sum_{n=1}^{N}\sum_{k=1}^{K}\mathbb{I}(m_{k,n} \equiv m'_{k,n}) \tag{12}$$

4.2.2. Image Quality

To evaluate the quality of watermarked images, we utilize two widely recognized metrics: FID and CLIP-Score.

To analyze the potential impact of watermark embedding on the model's performance, we conduct a statistical evaluation using a t-test. Specifically, we define the null hypothesis (H_0) and the alternative hypothesis (H_1) as $H_0 : \mu_s = \mu_0$ and $H_1 : \mu_s \neq \mu_0$, where μ_s and μ_0 represent the average FID or CLIP-Score for multiple sets of watermarked and watermark-free images, respectively. The t-test is used to determine whether the watermark embedding introduces statistically significant performance differences. A smaller t-value suggests a higher likelihood of accepting H_0, indicating no significant performance impact. Conversely, if the t-value exceeds a predefined threshold, H_0 is rejected, suggesting that watermark embedding may influence the model's performance.

4.3. Comparison with Baselines

4.3.1. Robustness Evaluation against Image Processing

We evaluated the robustness of various watermarking methods by extracting watermarks from 1,000 generated images under different image distortions.

As shown in Tables 3 and 4, the former reports the detection accuracy, while the latter reports the bit accuracy. The attacks labeled A–I correspond to those listed in Table 2. The results demonstrate that our method consistently achieved the highest watermark extraction accuracy across all distortion scenarios. Notably, in the Gaussian noise scenario, our method attained an average detection accuracy of 99.8%, surpassing the best-performing baseline, Gaussian Shading (Yang et al., 2024), by 23.2%. These results can be attributed to our fine-tuned LDM encoder, which demonstrates strong adaptability to both noise injection and denoising attacks. It effectively preserves the fidelity of the latent representations during encoding and decoding, ensuring that the recovered latent variables remain closely aligned with the originals. This significantly enhances the accuracy of watermark extraction.

4.3.2. Robustness Evaluation against VAE-Based Regeneration Attacks

In our experiments, to evaluate the robustness of watermarking methods against VAE-based regeneration attacks, we used two VAE-based image compression models from the compressAI library: Bmshj2018 (Ballé et al., 2018) and Cheng2020 (Cheng et al., 2020), referred to as Regen-VAE-A and Regen-VAE-B, respectively. The image quality factors were set to [1, 2, 3, 4, 5], where a lower value indicates stronger compression. The average results are shown in Tables 3 and 4. Several post-generation watermarking techniques fail after such attacks. While recent latent watermarking methods can resist the compression effects of these two VAE models to some extent, their performance is still inferior to the proposed method.

4.3.3. Robustness Evaluation against Diffusion-Based Regeneration Attacks

Diffusion-based regeneration attacks (Zhao et al., 2024) introduce noise and apply denoising processes to watermarked images using diffusion models, aiming to disrupt watermark information. Research indicates that most existing generative model watermarks exhibit poor robustness against diffusion-based attacks. To evaluate the performance of our method under such attacks, we selected five different diffusion step sizes: 300, 400, 500, 600, and 700. As the diffusion step size increases, the deviation between the image and the original image becomes larger, typically leading to performance degradation. As shown in Fig. 4, our method demonstrates the strongest resistance to this type of attack, which can be attributed to our grouping and rearrangement strategy. This strategy independently handles elements with small absolute values in the latent variables, effectively mitigating the impact of image deviations caused by the diffusion attack. In contrast, other post-generation watermarking methods fail to resist such attacks, causing detection accuracy to approach 0%. For Stable Signature (Fernandez et al., 2023), the watermark is completely erased during the attack process due to the use of a non-watermark decoder.

4.3.4. Image Quality

To quantitatively evaluate the quality of watermarked images, we calculated the FID and CLIP-Score of various watermarking methods relative to the baseline model. For

Table 3
Performance comparison of detection accuracy across methods under varying attacks

Method	A	B	C	D	E	F	G	H	I
DwtDct (Cox et al., 2007)	0.074	0.009	0.001	0.075	0.013	0.321	0.000	0.000	0.000
DwtDctSvd (Cox et al., 2007)	0.782	0.572	0.326	0.179	0.000	0.811	0.293	0.153	0.001
RivaGAN (Zhang et al., 2019)	0.483	0.834	0.213	0.102	0.058	0.744	0.061	0.017	0.000
SS (Fernandez et al., 2023)	0.192	0.555	0.000	0.065	0.004	0.432	0.267	0.221	0.000
Tree-Ring (Wen et al., 2023)	0.999	0.999	0.999	0.632	0.975	1.000	0.993	0.996	0.832
GS (Yang et al., 2024)	0.994	1.000	0.989	0.766	0.978	0.999	0.999	0.998	0.982
LW (Meng et al., 2025)	0.981	1.000	0.763	0.376	0.492	0.997	0.998	0.992	0.652
LaWa (Rezaei et al., 2024)	0.481	0.842	0.202	0.230	0.148	0.798	0.521	0.441	0.000
RingID (Ci et al., 2024)	0.990	0.978	0.960	0.537	0.894	0.991	0.963	0.969	0.728
PRC (Gunn et al., 2025)	0.492	0.791	0.314	0.083	0.073	0.795	0.574	0.634	0.009
T2SMark (Yang et al., 2025)	0.944	0.998	0.805	0.604	0.891	0.982	0.993	0.994	0.858
Proposed	**1.000**	**1.000**	**0.999**	**0.998**	**1.000**	**1.000**	**1.000**	**1.000**	**0.990**

Table 4
Performance comparison of bit accuracy across methods under varying attacks

Method	A	B	C	D	E	F	G	H	I
DwtDct (Cox et al., 2007)	0.5332	0.5166	0.5053	0.5206	0.5142	0.6120	0.5029	0.5013	0.5005
DwtDctSvd (Cox et al., 2007)	0.8047	0.7247	0.6357	0.5698	0.5037	0.8854	0.6082	0.5703	0.5016
RivaGAN (Zhang et al., 2019)	0.8459	0.8564	0.6837	0.6827	0.7044	0.9064	0.6459	0.6077	0.5427
SS (Fernandez et al., 2023)	0.6088	0.8121	0.4320	0.5702	0.5670	0.7399	0.7177	0.7021	0.4602
GS (Yang et al., 2024)	0.9445	0.9860	0.8957	0.7833	0.8685	0.9664	0.9782	0.9773	0.8442
LW (Meng et al., 2025)	0.8879	0.9648	0.8013	0.6473	0.6669	0.9480	0.9367	0.9371	0.6986
LaWa (Rezaei et al., 2024)	0.7994	0.9424	0.6477	0.6000	0.6165	0.9080	0.8233	0.7845	0.4676
PRC (Gunn et al., 2025)	0.7461	0.8957	0.6571	0.5418	0.5367	0.8978	0.7871	0.817	0.5043
T2SMark (Yang et al., 2025)	0.9432	0.9920	0.8656	0.7403	0.8683	0.9722	0.9807	0.9852	0.8441
Proposed	**0.9874**	**0.9941**	**0.9454**	**0.9597**	**0.9978**	**0.9809**	**0.9866**	**0.9854**	**0.8709**

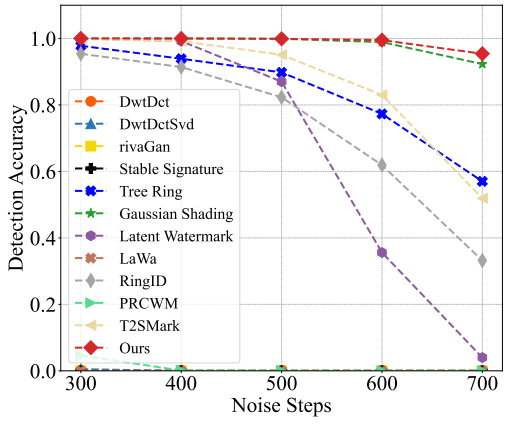


Fig. 4: Detection accuracy of watermarking methods under diffusion-based regeneration attacks.

FID, 10 sets of 5,000 images were generated from the COCO dataset. For CLIP-Score, 10 sets of 1,000 images were generated using Stable-Diffusion-Prompt. A t-test was conducted to measure statistical significance.

As demonstrated by the results in brackets in Table 5, the t-values for TraceMark-LDM are all below the threshold of 2.101 specified in (Yang et al., 2024). This indicates that TraceMark-LDM has a negligible impact on the quality and semantic consistency of synthesized images during watermark embedding. The majority of the remaining methods exceeded the threshold in at least one metric, indicating stronger interference with the generation process. For the CLIP-Score, our method achieved the best performance, showing the smallest deviation from the baseline. Concerning FID, our performance is comparable to the state-of-the-art methods. Additionally, we observed that some post-generation methods consistently resulted in lower FID scores. This phenomenon may be attributed to the watermarking process smoothing the texture of images, which can lead to visually more coherent results.

Fig. 5 shows visualization examples of different methods. Both TraceMark-LDM and initial-noise-based watermarking methods (Yang et al., 2024; Ci et al., 2024; Wen

Table 5
Comparison of image quality and embedding efficiency

Method	Image Quality		Embedding Time (ms)
	FID (t-value ↓)	CLIP-Score (t-value ↓)	
Stable Diffusion	$24.90_{.02}$	$0.3647_{.0011}$	-
DwtDct (Cox et al., 2007)	$24.76_{.02}$ (2.189)	$0.3629_{.0011}$ (3.608)	21.6
DwtDctSvd (Cox et al., 2007)	$24.27_{.02}$ (9.559)	$0.3622_{.0010}$ (5.059)	47.6
RivaGAN (Zhang et al., 2019)	$24.02_{.02}$ (13.81)	$0.3624_{.0011}$ (4.629)	410.6
SS (Fernandez et al., 2023)	$25.49_{.10}$ (10.03)	$0.3635_{.0011}$ (2.347)	**0.0**
Tree-Ring (Wen et al., 2023)	$25.34_{.02}$ (6.398)	$0.3631_{.0007}$ (3.632)	0.4
GS (Yang et al., 2024)	$24.88_{.07}$ (**0.169**)	$0.3645_{.0006}$ (**0.563**)	778.4
LW (Meng et al., 2025)	$27.12_{.16}$ (32.51)	$0.3583_{.0009}$ (13.99)	3.2
LaWa (Rezaei et al., 2024)	$24.45_{.23}$ (5.161)	$0.3617_{.0011}$ (6.692)	58.3
RingID (Ci et al., 2024)	$25.89_{.11}$ (17.53)	$0.3576_{.0007}$ (16.59)	0.6
PRCW (Gunn et al., 2025)	$24.85_{.14}$(**0.768**)	$0.3645_{.0010}$ (**0.425**)	5.2
T2SMark (Yang et al., 2025)	$24.85_{.16}$ (**0.752**)	$0.3640_{.0008}$ (**1.568**)	1.3
Proposed	$24.96_{.03}$ (**0.773**)	$0.3649_{.0010}$ (**0.372**)	15.4

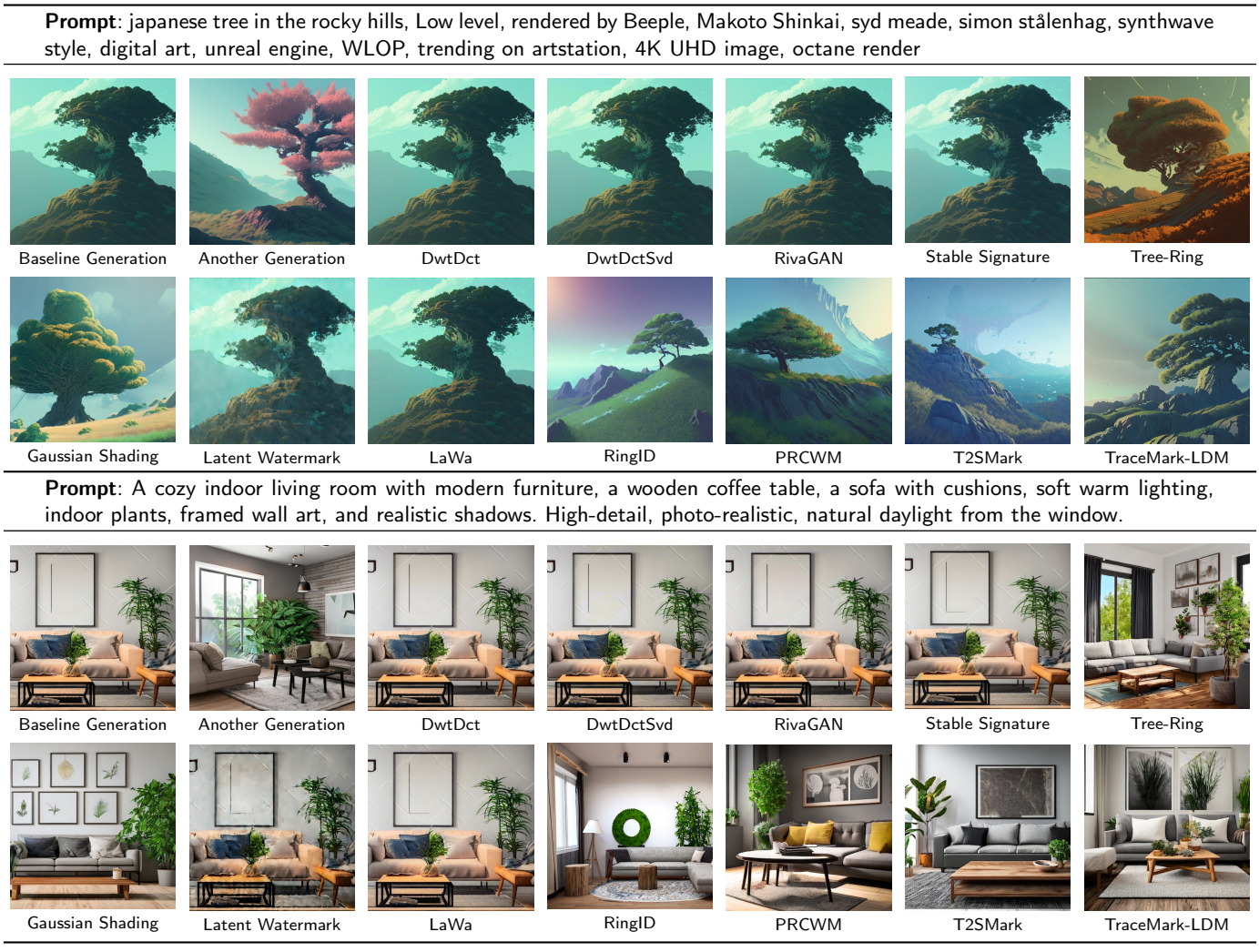


Fig. 5: Examples of different watermarking methods applied to images generated from various prompts. **Baseline Generation** refers to the standard generated image without any watermark, serving as the reference baseline. **Another Generation** refers to another independent image generated with the same prompt, showcasing the inherent diversity of the model.

Table 6
Ablation study of different components of our proposed method under various adversarial attacks

Attack	Methods			
	Ours	w/o FT, with GR	w/o FT, w/o S	w/o FT, with R
MedianFilter	1.000/0.9874	0.997/0.9579	0.996/0.9546	0.993/0.9448
JPEG	1.000/0.9941	0.999/0.9906	1.000/0.9900	0.999/0.9853
GauBlur	0.999/0.9454	0.994/0.9178	0.993/0.9137	0.987/0.8978
GauNoise	0.998/0.9597	0.810/0.8022	0.785/0.7939	0.749/0.7763
S&P Noise	1.000/0.9978	0.992/0.8957	0.989/0.8854	0.975/0.8603
Resize	1.000/0.9809	1.000/0.9739	1.000/0.9722	0.998/0.9670

et al., 2023) produce noticeable visual differences compared to the non-watermarked outputs. These discrepancies arise because the watermark is embedded at the noise level, rather than in the image or feature space. However, it is important to note that in diffusion models, different random noise seeds guided by the same prompt can still produce images that are semantically similar but visually different. Therefore, not all observed differences should be attributed solely to the watermark embedding process. Finally, the Latent Watermark (Meng et al., 2025) exhibits noticeable visual distortion as the watermark length increases, highlighting the difficulty of maintaining both robustness and image quality in this setting.

4.3.5. Time Efficiency

Watermarking methods for diffusion models should minimize the impact on generation efficiency. We compared the embedding time of our method with others. As shown in Table 5, fine-tuning-based methods do not introduce additional embedding time during inference, but they incur substantial computational and time costs during training. On the other hand, while Gaussian Shading (Yang et al., 2024) maintains high image quality, its watermark embedding takes approximately 778.4 ms, significantly reducing efficiency and potentially hindering user experience in real-world applications. In contrast, TraceMark-LDM requires only 15.4 ms, which constitutes a negligible processing overhead.

4.4. Ablation Studies

We conducted a series of ablation studies to validate the significance of fine-tuning and group rearrangement, and to evaluate the impact of various parameters on our method, including watermark length, inversion step, sampling method, and guidance scale.

4.4.1. Effectiveness of Fine-Tuning

To verify the effectiveness of the fine-tuned encoder, we applied the same attack scenarios to images generated by the unfine-tuned version. As shown in Table 6, "w/o FT, with GR" denotes the variant without fine-tuning. The data before and after the slash represent the detection and bit accuracy, respectively. The attacks labeled A–F correspond to those listed in Table 2. The results show that even without fine-tuning, our method outperforms existing baselines. Nevertheless, the fine-tuned version achieves higher detection and bit accuracy, particularly under Gaussian noise, with improvements of 18.8% and 15.75%, respectively. These findings suggest that fine-tuning enables the encoder to better adapt to noise perturbations introduced by adversarial attacks, thereby improving the alignment between the embedded and extracted watermark.

4.4.2. Effectiveness of Group Rearrangement

To evaluate the effectiveness of the proposed Group Rearrangement (GR) strategy for small-magnitude elements, we conducted ablation tests using different handling strategies. As shown in Table 6, "with GR" applies GR to these elements; "w/o S" excludes them from embedding; and "with R" applies a basic Rearrangement (R) strategy. The results show that removing these elements outperforms directly applying rearrangement to all elements. Due to their near-zero values, these elements are highly sensitive to distortion and prone to sign flipping, which compromises the voting process and reduces watermark accuracy. Furthermore, by selectively utilizing these elements via the Group Rearrangement strategy, watermark extraction accuracy consistently improved across various distortion scenarios. Although the gains are modest, the results clearly demonstrate the reliability and effectiveness of the proposed strategy.

4.4.3. Watermark Length

We evaluated bit accuracy under different watermark lengths, with results summarized in Table 7. As the watermark length increases, bit accuracy gradually declines. This is due to the dimensionality of latent variables remains constant under fixed image resolution. In this case longer watermarks reduce repetition during embedding, yielding fewer groups for majority voting during extraction and thus lower accuracy. Despite this, our method maintains strong performance: under clean conditions, it achieves 96.8% accuracy even at 4096 bits. Although the accuracy is sometimes slightly lower than T2SMark, this is attributed to the trade-off between robustness and accuracy introduced by the fine-tuned encoder. Notably, we observe that T2SMark degrades much more rapidly as the watermark length increases in

Table 7
Bit accuracy with different watermark lengths

Method		Watermark Length					
		128	256	512	1024	2048	4096
GS (Yang et al., 2024)	Clean	1.0000	1.0000	0.9999	0.9985	0.9852	0.9406
	Adversarial	0.9551	0.9225	0.8799	0.8282	0.7690	0.7101
PRC (Gunn et al., 2025)	Clean	1.0000	1.0000	1.0000	0.9929	0.5593	0.5017
	Adversarial	0.7200	0.7042	0.6684	0.6225	0.5000	0.5000
T2SMark (Yang et al., 2025)	Clean	1.0000	1.0000	0.9999	0.9996	0.9964	0.9808
	Adversarial	0.9458	0.9235	0.8910	0.8459	0.7949	0.7392
Proposed	Clean	1.0000	1.0000	0.9998	0.9983	0.9926	0.9680
	Adversarial	0.9967	0.9886	0.9671	0.9327	0.8807	0.8188

distorted conditions. In contrast, our method consistently outperforms all baselines across the A–F adversarial settings (Table 2), demonstrating substantially stronger robustness.

Furthermore, embedding high-capacity watermarks (e.g., 4096 bits) does not degrade image quality or prompt alignment. This is attributed to our method's reliance on the rearrangement mechanism of independent and identically distributed (i.i.d.) Gaussian samples. As proven in Eq. 3, the rearranged noise vector $\mathbf{z}_T^w$ strictly retains the standard Gaussian distribution regardless of the watermark length k. Consequently, the watermarked input remains statistically indistinguishable from natural noise to the LDMs. Correspondingly, high-capacity watermarking only comes at the expense of robustness due to the reduced redundancy for majority voting.

4.4.4. Inversion Step

Typically, inversion is most effective when its step size matches that of the original inference process. However, in practice, the exact inference step size used during watermark generation is often unknown. We evaluated various combinations of inversion and inference step sizes, with results summarized in Table 8. The results show that our method is robust to the choice of inversion step size. Even with a small inversion step size of 10, bit accuracy remains close to 100%. Furthermore, with an inversion step size of 50, watermark information can be accurately extracted across all tested inference step sizes under clean conditions.

4.4.5. Sampling Method

To evaluate the impact of different sampling methods on bit accuracy, we tested five approaches under both clean and distorted conditions.

As shown in Table 9, where A–F correspond to the attacks listed in Table 2 and the subscripts indicate the severity of each attack, the results demonstrate that all sampling methods achieved satisfactory watermark extraction performance, offering users greater flexibility in practical deployment and enhancing the overall adaptability of our approach. Notably, DPMSolver attained an average bit accuracy of 98.86% across various distortion scenarios, demonstrating strong robustness and resistance to interference. These results confirm that our method is compatible with multiple sampling strategies and can maintain stable extraction performance across diverse application environments.

Table 8
Bit accuracy with different inference and inversion steps

Inference Step	Inversion Step			
	10	25	50	100
10	0.9999	0.9999	1.0000	1.0000
25	1.0000	1.0000	1.0000	1.0000
50	1.0000	1.0000	1.0000	1.0000
100	1.0000	1.0000	1.0000	1.0000

Table 9
Bit accuracy across different sampling methods

Attack	Sampling Methods				
	DDIM	UniPC	PNDM	DEIS	DPMSolver
Clean	1.0000	1.0000	1.0000	1.0000	1.0000
$A_{\text{Kernel Size=11}}$	0.9928	0.9914	0.9945	0.9936	0.9944
$B_{\text{QF=50}}$	0.9979	0.9976	0.9985	0.9982	0.9985
$C_{\text{Radius=6}}$	0.9598	0.9549	0.9663	0.9626	0.9660
$D_{\text{Std=0.15}}$	0.9660	0.9640	0.9720	0.9704	0.9738
$E_{\text{Prob=0.2}}$	0.9982	0.9977	0.9987	0.9984	0.9987
$F_{\text{Factor=0.5}}$	0.9999	0.9999	1.0000	0.9999	0.9999
AvgAdv	0.9858	0.9843	0.9883	0.9872	0.9886

4.4.6. Guidance Scale

By adjusting the guidance scale, a balance can be achieved between image quality and generation diversity. To verify that users can still successfully extract watermark

Table 10
Bit accuracy across different guidance scales.

Attack	Guidance Scale				
	2.0	6.0	10.0	14.0	18.0
Clean	1.0000	1.0000	1.0000	0.9998	0.9991
$A_{\text{Kernel Size=11}}$	0.9993	0.9959	0.9909	0.9835	0.9735
$B_{\text{QF=50}}$	0.9999	0.9992	0.9970	0.9943	0.9899
$C_{\text{Radius=6}}$	0.9895	0.9723	0.9539	0.9336	0.9124
$D_{\text{Std=0.15}}$	0.9969	0.9810	0.9610	0.9406	0.9232
$E_{\text{Prob=0.2}}$	0.9999	0.9991	0.9972	0.9936	0.9894
$F_{\text{Factor=0.5}}$	1.0000	1.0000	0.9998	0.9993	0.9982
AvgAdv	0.9971	0.9895	0.9800	0.9691	0.9577

Table 11
Ablation study on the weight coefficient λ in the loss function.

Attack	Weight Coefficient (λ)			
	0	0.5	1	2
Clean	1.0000	1.0000	1.0000	1.0000
MedianFilter	0.9934	0.9935	0.9944	0.9854
JPEG	0.9987	0.9986	0.9985	0.9982
GauBlur	0.9533	0.9647	0.9660	0.9115
GauNoise	0.9209	0.9741	0.9738	0.9674
S&P Noise	0.9593	0.9981	0.9987	0.9911
Resize	0.9999	0.9999	0.9999	0.9999
AvgAdv	0.9709	0.9881	0.9886	0.9756

information under different guidance scale settings, we selected five evenly spaced values within the range of 2 to 18 for testing. It is worth noting that a larger guidance scale forces the generated image to adhere more strictly to the given prompt, while our inversion process adopts an empty prompt. This discrepancy increases the deviation when recovering the initial noise, which may affect watermark extraction accuracy to some extent. Nevertheless, as shown in Table 10, our method consistently maintains a high level of bit accuracy across all guidance scale values. These results further demonstrate the robustness and broad applicability of our approach.

4.4.7. Impact of the Weight Coefficient λ

We conduct an ablation study with $\lambda \in \{0, 0.5, 1, 2\}$ to analyze the trade-off between the latent-variable mean squared error (MSE) loss and the perceptual loss of the reconstructed images. The detailed results are reported in Table 11.

The baseline model trained only with the MSE loss ($\lambda = 0$) achieves an average accuracy of 97.09% but shows limited robustness to Gaussian and salt-and-pepper noise. This is mainly because MSE focuses on pixel-level local consistency and is prone to overfitting to fine-grained noise, making it difficult to ensure global consistency in the reconstructed results. When λ is increased to 1, and the perceptual constraint is introduced, the average accuracy under Gaussian noise and salt-and-pepper noise increases significantly to 97.38% and 99.87%, respectively. This indicates that perceptual loss can align high-level semantics and overall structure, effectively mitigating noise interference in pixel-space and improving global consistency.

However, when λ is further increased to 2, the average accuracy under adversarial settings decreases to 97.56%. This is because an excessively large weight makes the perceptual loss dominate the optimization process, causing the model to prioritize perceptual similarity and weakening the precise alignment of latent variables in the latent space. With the MSE constraint relatively weakened, the inversion solution may deviate from the original latent representation, thereby degrading the final performance. Therefore, we choose $\lambda = 1$ as the configuration with the best overall performance.

Table 12
Comparison of performance under different flip probabilities p.

p	0	0.1	0.2	0.3	0.4
GS	0.9920	0.9826	0.9592	0.9410	0.7972
Proposed	0.9955	0.9875	0.9707	0.9605	0.8179

4.5. Additional Discussions

4.5.1. Inversion Attack

In real-world scenarios, attackers can use a diffusion model to invert a watermarked image, recover the noise vector, randomly flip the signs of certain elements, and regenerate the image to remove the watermark. To assess this attack, we conducted tests, measuring attack intensity by the proportion of inverted elements (0 to 0.4). Results show that as attack intensity increases, our method significantly outperforms Gaussian Shading (Yang et al., 2024). The experimental results in Table 12 indicate that as the attack intensity increases, our method demonstrates a progressively significant advantage over Gaussian Shading. The enhanced adversarial robustness stems from the grouping strategy for low absolute value elements, where intra-group summation inherently compensates for sign-flipping perturbations. This mechanism effectively mitigates attack-induced distortions while preserving algorithmic stability.

4.5.2. Ethical Implications and Responsible Use

Watermarking for AIGC serves as a sociotechnical safeguard that follows value-sensitive design (VSD) principles Friedman et al. (2013), which emphasize that it is essential to consider not only the practicality of the technology but also values such as human well-being, privacy, security, and fairness during the design phase. Watermarking technology reduces the harm of misinformation on public perception and social order by maintaining content credibility, thereby safeguarding human well-being. At the same time,

it leverages source tracking to protect user data privacy and personal information security. Additionally, through clear copyright identification, it addresses the issue of "unclear authorship," creating a fair environment for the distribution of rights for creators of all sizes and preventing ethical imbalances under the benefits of technology.

To support the responsible deployment of TraceMark-LDM, we recommend avoiding the embedding of personal identifiers in payloads, maintain segregated and regularly rotated keys. Additionally, high-stakes decisions should require human review, and users should be informed of authenticity indicators whenever feasible. Furthermore, we periodically assess the system's robustness against regeneration and forgery attacks to ensure timely fine-tuning of the encoder, thereby maintaining the high reliability of TraceMark.

4.5.3. Limitation

The proposed method demonstrates certain compatibility constraints with stochastic differential equation (SDE)-based samplers. This limitation primarily stems from the irreversible nature of stochastic noise introduced during SDE sampling, which disrupts the DDIM inversion process and consequently hinders accurate recovery of the initial watermark-embedded noise. For practical implementation and deployment, we strongly recommend prioritizing deterministic samplers based on ordinary differential equations (ODEs), as these not only align with our method's requirements but also benefit from extensive support in mainstream frameworks.

5. Conclusion and Future Work

In this study, we propose TraceMark-LDM, a general and efficient watermarking approach specially designed for LDMs. Unlike traditional watermarking schemes, our method does not directly embed the watermark into images. Instead, it uses the watermark to guide the rearrangement of the initial noise in the LDMs. Experimental results demonstrate that TraceMark-LDM effectively enables the tracing and detection of generated content, enhancing the supervision of synthesized images and the trustworthiness of digital data. Furthermore, the FID and CLIP-Score results achieved by TraceMark-LDM are highly comparable to those of other SOTA models.

In the future, the challenge of provable video generation is an intriguing one that could be usefully explored in our further research. We plan to adapt and refine our methods for application in the video generation models, such as the Diffusion Transformer (DiT) behind SORA.

CRediT authorship contribution statement

Wenhao Luo: Methodology, Software, Investigation, Writing original draft. **Zhangyi Shen**: Methodology, Conceptualization, Validation, Writing original draft. **Ye Yao**: Methodology, Software, Validation. **Feng Ding**: Methodology, Supervision, Review and editing. **Guopu Zhu**: Conceptualization, Supervision, Review and editing. **Weizhi Meng**: Conceptualization, Supervision, Review and editing.

Funding

This work was supported by the National Natural Science Foundation of China under Grant Number 62471167. The authors acknowledge the Supercomputing Center of Hangzhou Dianzi University for providing computing resources.

References

Badran, E.F., Sharkas, M.A., Attallah, O.A., 2009. Multiple watermark embedding scheme in wavelet-spatial domains based on roi of medical images, in: 2009 National Radio Science Conference, pp. 1–8.

Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N., 2018. Variational image compression with a scale hyperprior, in: 6th International Conference on Learning Representations, ICLR.

Bui, T., Yu, N., Collomosse, J., 2022. Repmix: Representation mixing for robust attribution of synthesized images, in: European Conference on Computer Vision, pp. 146–163.

Cheng, Z., Sun, H., Takeuchi, M., Katto, J., 2020. Learned image compression with discretized gaussian mixture likelihoods and attention modules. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition , 7936–7945.

Ci, H., Yang, P., Song, Y., Shou, M.Z., 2024. Ringid: Rethinking tree-ring watermarking for enhanced multi-key identification, in: European Conference on Computer Vision, pp. 338–354.

Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T., 2007. Digital watermarking and steganography. Morgan kaufmann.

Cui, Y., Ren, J., Xu, H., He, P., Liu, H., Sun, L., Xing, Y., Tang, J., 2025. Diffusionshield: A watermark for data copyright protection against generative diffusion models. ACM SIGKDD Explorations Newsletter 26, 60–75.

Dhariwal, P., Nichol, A., 2021. Diffusion models beat gans on image synthesis. Advances in neural information processing systems 34, 8780–8794.

Fan, B., Hu, S., Ding, F., 2024. Synthesizing black-box anti-forensics deepfakes with high visual quality, in: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4545–4549.

Fan, G., Pan, Z., Zhou, Q., Gao, X., Zhang, X., 2021. Multiple histogram based adaptive pairwise prediction-error modification for efficient reversible image watermarking. Information Sciences 581, 515–535.

Fang, H., Jia, Z., Qiu, Y., Zhang, J., Zhang, W., Chang, E.C., 2022. Deend: decoder-driven watermarking network. IEEE Transactions on Multimedia 25, 7571–7581.

Fang, H., Zhou, H., Ma, Z., Zhang, W., Yu, N., 2019. A robust image watermarking scheme in dct domain based on adaptive texture direction quantization. Multimedia Tools and Applications 78, 8075–8089.

Feng, W., Zhou, W., He, J., Zhang, J., Wei, T., Li, G., Zhang, T., Zhang, W., Yu, N., 2024. Aqualora: toward white-box protection for customized stable diffusion models via watermark lora, in: Proceedings of the 41st International Conference on Machine Learning, pp. 13423–13444.

Fernandez, P., Couairon, G., Jégou, H., Douze, M., Furon, T., 2023. The stable signature: Rooting watermarks in latent diffusion models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 22466–22477.

Friedman, B., Kahn Jr, P.H., Borning, A., Huldtgren, A., 2013. Value sensitive design and information systems, in: Early engagement and new technologies: Opening up the laboratory. Springer, pp. 55–95.

Frolov, S., Hinz, T., Raue, F., Hees, J., Dengel, A., 2021. Adversarial text-to-image synthesis: A review. Neural Networks 144, 187–209.

Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B., 2022. Vector quantized diffusion model for text-to-image synthesis, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10696–10706.

Gunn, S., Zhao, X., Song, D., 2025. An undetectable watermark for generative image models, in: The Thirteenth International Conference on Learning Representations.

Guo, D., Chen, H., Wu, R., Wang, Y., 2023. Aigc challenges and opportunities related to public safety: a case study of chatgpt. Journal of Safety Science and Resilience 4, 329–339.

He, W., Cai, Z., Wang, Y., 2020. High-fidelity reversible image watermarking based on effective prediction error-pairs modification. IEEE Transactions on Multimedia 23, 52–63.

Ho, J., Jain, A., Abbeel, P., 2020. Denoising diffusion probabilistic models. Advances in neural information processing systems 33, 6840–6851.

Ho, J., Salimans, T., 2022. Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 .

Hu, X., Li, S., Ying, Q., Peng, W., Zhang, X., Qian, Z., 2024. Establishing robust generative image steganography via popular stable diffusion. IEEE Transactions on Information Forensics and Security 19, 8094–8108.

Huan, W., Li, S., Qian, Z., Zhang, X., 2021. Exploring stable coefficients on joint sub-bands for robust video watermarking in dt cwt domain. IEEE Transactions on Circuits and Systems for Video Technology 32, 1955–1965.

Jia, Z., Fang, H., Zhang, W., 2021. 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, pp. 41–49.

Kang, X., Huang, J., Shi, Y.Q., Lin, Y., 2003. A dwt-dft composite watermarking scheme robust to both affine transform and jpeg compression. IEEE transactions on circuits and systems for video technology 13, 776–786.

Kingma, D.P., Welling, M., 2014. Auto-encoding variational bayes, in: 2nd International Conference on Learning Representations, ICLR.

Li, Y., Liao, X., Wu, X., 2024. Screen-shooting resistant watermarking with grayscale deviation simulation. IEEE Transactions on Multimedia 26, 10908–10923.

Liu, A.A., Su, Y., Wang, L., Li, B., Qian, Z., Zhang, W., Zhou, L., Zhang, X., Zhang, Y., Huang, J., Yu, N., 2024. Review on the progress of the aigc visual content generation and traceability. Journal of Image and Graphics 29, 1535–1554.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J., 2022. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps, in: Advances in Neural Information Processing Systems, pp. 5775–5787.

Luo, T., Hu, R., He, Z., Jiang, G., Xu, H., Song, Y., Chang, C.C., 2026. Diffw: Multi-encoder based on conditional diffusion model for robust image watermarking. IEEE Transactions on Multimedia 28, 837–852.

Luo, T., Wu, J., He, Z., Xu, H., Jiang, G., Chang, C.C., 2024. Wformer: A transformer-based soft fusion model for robust image watermarking. IEEE transactions on emerging topics in computational intelligence 8, 4179–4196.

Ma, R., Guo, M., Hou, Y., Yang, F., Li, Y., Jia, H., Xie, X., 2022. Towards blind watermarking: Combining invertible and non-invertible mechanisms, in: Proceedings of the 30th ACM International Conference on Multimedia, pp. 1532–1542.

Meng, Z., Peng, B., Dong, J., 2025. Latent watermark: Inject and detect watermarks in latent diffusion space. IEEE Transactions on Multimedia 27, 3399–3410.

Mohammed, G.N., Yasin, A., Zeki, A.M., 2014. Robust image watermarking based on dual intermediate significant bit (disb), in: 2014 6th International Conference on Computer Science and Information Technology (CSIT), pp. 18–22.

Nichol, A.Q., Dhariwal, P., 2021. Improved denoising diffusion probabilistic models, in: International conference on machine learning, pp. 8162–8171.

Qiao, T., Zhang, J., Xu, D., Tao, D., 2019. Mirrorgan: Learning text-to-image generation by redescription, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 1505–1514.

Qin, C., Ji, P., Zhang, X., Dong, J., Wang, J., 2017. Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy. Signal processing 138, 280–293.

Qin, C., Li, X., Zhang, Z., Li, F., Zhang, X., Feng, G., 2024. Print-camera resistant image watermarking with deep noise simulation and constrained learning. IEEE Transactions on Multimedia 26, 2164–2177.

Rezaei, A., Akbari, M., Alvar, S.R., Fatemi, A., Zhang, Y., 2024. Lawa: Using latent space for in-generation image watermarking, in: European Conference on Computer Vision, pp. 118–136.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B., 2022. High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10684–10695.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S., 2015. Deep unsupervised learning using nonequilibrium thermodynamics, in: International conference on machine learning, pp. 2256–2265.

Song, J., Meng, C., Ermon, S., 2021a. Denoising diffusion implicit models, in: 9th International Conference on Learning Representations, ICLR.

Song, Y., Ermon, S., 2019. Generative modeling by estimating gradients of the data distribution, in: Advances in neural information processing systems.

Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B., 2021b. Score-based generative modeling through stochastic differential equations, in: 9th International Conference on Learning Representations, ICLR.

Van Schyndel, R.G., Tirkel, A.Z., Osborne, C.F., 1994. A digital watermark, in: Proceedings of 1st international conference on image processing, pp. 86–90.

Wang, Y., Ying, Q., Sun, Y., Qian, Z., Zhang, X., 2022. A dtcwt-svd based video watermarking resistant to frame rate conversion, in: 2022 International Conference on Culture-Oriented Science and Technology (CoST), pp. 36–40.

Wen, Y., Kirchenbauer, J., Geiping, J., Goldstein, T., 2023. Tree-rings watermarks: Invisible fingerprints for diffusion images, in: Advances in Neural Information Processing Systems, pp. 58047–58063.

Xiong, C., Qin, C., Feng, G., Zhang, X., 2023. Flexible and secure watermarking for latent diffusion model, in: Proceedings of the 31st ACM International Conference on Multimedia, pp. 1668–1676.

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., He, X., 2018. Attngan: Fine-grained text to image generation with attentional generative adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1316–1324.

Yang, J., Fang, H., Zhang, W., Yu, N., Chen, K., 2025. T2smark: Balancing robustness and diversity in noise-as-watermark for diffusion models, in: The Thirty-ninth Annual Conference on Neural Information Processing Systems.

Yang, Z., Zeng, K., Chen, K., Fang, H., Zhang, W., Yu, N., 2024. Gaussian shading: Provable performance-lossless image watermarking for diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12162–12171.

Yin, Z., Yin, H., Zhang, X., 2022. Neural network fragile watermarking with no model performance degradation, in: 2022 IEEE International Conference on Image Processing (ICIP), pp. 3958–3962.

Yuan, Z., Li, L., Wang, Z., Zhang, X., 2024. Watermarking for stable diffusion models. IEEE Internet of Things Journal 11, 35238–35249.

Zhang, K.A., Xu, L., Cuesta-Infante, A., Veeramachaneni, K., 2019. Robust invisible video watermarking with attention. ArXiv abs/1909.01285.

Zhang, X., Li, R., Yu, J., Xu, Y., Li, W., Zhang, J., 2024. Editguard: Versatile image watermarking for tamper localization and copyright protection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 11964–11974.

Zhao, X., Zhang, K., Su, Z., Vasan, S., Grishchenko, I., Kruegel, C., Vigna, G., Wang, Y.X., Li, L., 2024. Invisible image watermarks are provably removable using generative ai. Advances in neural information processing systems 37, 8643–8672.

Zhong, X., Huang, P.C., Mastorakis, S., Shih, F.Y., 2020. An automated and robust image watermarking scheme based on deep neural networks.

IEEE Transactions on Multimedia 23, 1951–1961.
Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L., 2018. Hidden: Hiding data with deep networks, in: Proceedings of the European conference on computer vision (ECCV), pp. 657–672.