A Watermark-Conditioned Diffusion Model for IP Protection

Rui Min[1], Sen Li[1], Hongyang Chen[2], and Minhao Cheng[3]

[1] Hong Kong University of Science and Technology
[2] Zhejiang Lab
[3] Pennsylvania State University
{rminaa,slien}@connect.ust.hk,
hongyang@zhejianglab.com, mmc7149@psu.edu

Abstract. The ethical need to protect AI-generated content has been a significant concern in recent years. While existing watermarking strategies have demonstrated success in detecting synthetic content (**detection**), there has been limited exploration in identifying the users responsible for generating these outputs from a single model (**owner identification**). In this paper, we focus on both practical scenarios and propose a unified watermarking framework for content copyright protection within the context of diffusion models. Specifically, we consider two parties: the model provider, who grants public access to a diffusion model via an API, and the users, who can solely query the model API and generate images in a black-box manner. Our task is to embed hidden information into the generated contents, which facilitates further detection and owner identification. To tackle this challenge, we propose a **Wa**termark-conditioned **Diff**usion model called WaDiff, which manipulates the watermark as a conditioned input and incorporates fingerprinting into the generation process. All the generative outputs from our WaDiff carry user-specific information, which can be recovered by an image extractor and further facilitate forensic identification. Extensive experiments are conducted on two popular diffusion models, and we demonstrate that our method is effective and robust in both the detection and owner identification tasks. Meanwhile, our watermarking framework only exerts a negligible impact on the original generation and is more stealthy and efficient in comparison to existing watermarking strategies. Our code is publicly available at `https://github.com/rmin2000/WaDiff`.

Keywords: Digital watermark · Diffusion model · IP protection

1 Introduction

The recent progress in diffusion models has significantly advanced the field of AI-generated content (AIGC). Notably, several popular public APIs, such as Stable Diffusion [27] and DALL·E 3 [2], have emerged, providing users with convenient access to create and personalize high-quality images. However, as these systems become more pervasive, the risk of malicious use and attacks increases. In particular, some users with malicious intent may exploit the powerful generation capabilities of these models to create photo-realistic images like deep fakes [25], which can then be disseminated for illegal purposes. Moreover, as generative models are excellent tools for creating and manipulating content, it is crucial to safeguard users' copyrights and intellectual property when using state-of-the-art generative models. By discerning the source of each user's generated output, we can ensure that legitimate users' contributions are protected and prevent unauthorized replication or use of their content.

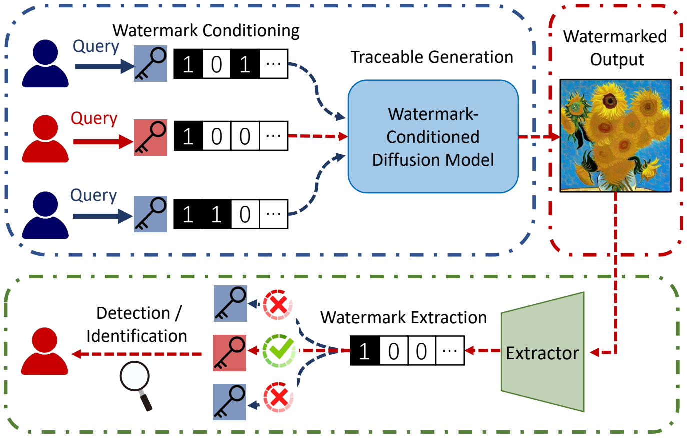

Fig. 1: Illustration of our proposed WaDiff. All users access the diffusion model by querying the public API and are assigned a unique watermark. The generation process is conditioned on the watermark, and each user's generated outputs would carry specific fingerprinting information which is further utilized to identify the owner of the generated image.

To enable the traceability of diffusion-generated images, a commonly employed strategy is to embed a unique fingerprint to contents generated by an individual user and then forensically identify the owner from the watermarked image. A line of previous works [1,12,26,29,45] has extensively investigated this approach within the realm of traditional multimedia copyright protection, which is commonly referred to as post-hoc watermark. Typically, these strategies involve embedding an imperceptible fingerprint into the generative content while leaving an identifiable trace that can be detected using a pre-designed mechanism. However, the post-hoc watermark requires additional computational costs for watermark injection and is more susceptible to circumvention. For instance, in the event of model leakage, attackers can easily detect and bypass the post-processing module. Recently, the Stable Signature [11] investigated the latent diffusion paradigm and proposed a method that incorporates watermarks into each latent decoder. However, solely fingerprinting the latent decoder limits their application scenario to the latent diffusion paradigm only and can be easily circumvented by retraining the latent decoder on a clean dataset. At the same time, since only a fixed watermark could be embedded into the latent decoder, every model has to be fine-tuned before being distributed to the users, making it hard to use in large-scale real-world systems. Considering both the scalability and effectiveness, we aim to investigate whether we can embed fingerprints during the generation process to incorporate user-specific watermarks without customized fine-tuning.

In this paper, we introduce WaDiff (as shown in Figure 1), a watermark-conditioned diffusion model, which incorporates the watermark as a conditioned input and generates images with unique fingerprints tailored to individual users. Unlike previous approaches [11, 36], which solely focus on fingerprinting the latent decoder in the diffusion model, our watermarking strategy is seamlessly integrated into the diffusion generation process. This makes our approach more general and applicable to other types of diffusion models [14] that do not include a latent decoder, while simultaneously eliminating the need for post-processing. Our WaDiff builds upon a pre-trained diffusion model, with slight modifications to the original input layers to accommodate the inclusion of watermark information by expanding the channels. Specifically, we first embed the watermark bits through a linear layer and then concatenate this projected vector with the original input to construct the watermark-conditioned input. We design a unified watermarking framework with two novel objective functions, named message retrieval loss and consistency loss. The message retrieval loss ensures the effective embedding of fingerprints into the generated content, allowing for successful retrieval of the embedded watermark. Meanwhile, the consistency loss ensures that the inclusion of the watermark has a negligible effect on the overall generation quality.

To this end, we evaluate our method on two popular open-source diffusion models and perform both detection and identification tasks. We also make detailed comparisons with widely used post-hoc watermarking strategies and Tree-Ring [34], a training-free fingerprinting framework for diffusion models. Experimental results demonstrate that our efficient watermarking strategy enables accurate detection and identification in a large-scale system with numerous users and remains robust across various data augmentations. Moreover, the generated images after watermarking maintain exceptional generation quality and are visually indistinguishable across different watermarks. Our contributions are organized as follows:

- Different from previous work, we propose a scalable watermarking strategy that efficiently integrates user-specific fingerprints into the diffusion generation process without the need for customized fine-tuning.
- We introduce WaDiff, a watermark-conditioned diffusion model, along with a unified watermarking framework. Unlike post-hoc fingerprinting, WaDiff manipulates the watermark as conditioned input and allows for effective watermarking while having a negligible impact on the generation quality.
- Extensive experiments demonstrate that our watermarking framework achieves precise and robust performance in detecting AI-generated content and identifying the source owner of generated images.

2 Related Work

2.1 Diffusion Models

Diffusion models have recently shown tremendous capability for high-quality image generation [14,27,28,30,31]. Depending on the space in which the generation process is performed, diffusion models can be divided into pixel-space [9,14] and latent-space diffusion models [27]. In pixel-space diffusion models, images are directly generated starting from sampled Gaussian noise [14]. To reduce the computational complexity, latent diffusion models have been proposed to first generate latent features from noise and then decode the latent features to images by VAE [27]. With the powerful generation capability, they have demonstrated amazing results on various computer vision tasks, such as text-to-image generation [23,28], sketch-to-image generation [22], text-guided image editing [23].

2.2 Image Watermarking

Image watermarking has been a broadly explored technique for decades. Traditional strategies typically start from a host image and inject watermark information directly in the spatial domain [26,29], or through certain domain transformations such as Discrete Cosine Transform (DCT) [1] and Discrete Wavelet Transform (DWT) [12]. With advancements in deep learning, researchers have also explored the use of deep-learning techniques to replace manually designed watermark patterns. For example, a line of works [19, 32, 40, 44, 45] leverages the capability of Deep Neural Networks (DNNs) to improve the stealthiness and robustness of the watermark. In addition to facilitating image protection, image watermarking techniques have been proved useful in various security challenges, such as model ownership verification [35, 41, 42], backdoor attack [17], dataset copyright protection [16], and forensic adversarial defense [4]. In our method, we leverage image watermarking to embed unique watermark information for individual users, thereby facilitating subsequent detection and identification.

2.3 Fingerprinting in Diffusion Models

In light of recent advancements in generative AI, researchers have been exploring watermarks to safeguard or regulate the usage of generative models [5, 7, 10, 15, 21, 37–39, 43]. In this study, we primarily focus on fingerprinting the generative content. The Stable Signature [11] first proposed a watermarking scheme for latent diffusion models [27] by fingerprinting a set of latent decoders. They then distributed these customized decoders to individual users for both detection and identification. Building upon this work, [36] improved the scalability by incorporating a message matrix into the latent decoder, allowing multiple watermarks to be carried within a single architecture. However, their methods only focus on manipulating the latent decoder while keeping the diffusion process intact, which makes their methods only applicable to latent-space diffusion models. Also, this leaves them vulnerable to attacks like simply retraining the fingerprinted latent decoder on a clean dataset. In a parallel work called Tree-Ring [34], researchers concealed the fingerprint within the frequency domain of the initial noisy vector. Detection is then performed by reversing the generative image back to the noisy vector and comparing it with the original noisy pattern. Despite its training-free nature, the detection process heavily relies on the time-consuming reversion process and poses challenges for identification among multiple users.

In contrast to previous studies, we propose the WaDiff along with a simple and unified watermarking framework for both detection and owner identification. Our method seamlessly integrates the fingerprinting process into the image generation process, which can be applied with various diffusion types and sampling schedules, resulting in enhanced stealthiness and compatibility.

3 Problem Setting

The model provider deploys a generation model and grants public access to m registered users. Considering the intellectual property (IP) protection, the model architecture and parameters are concealed from users, which means only black-box access for users is permitted and all the internal information remains encrypted. Note that the black-box scenario is common in practice, and has been widely adopted in current generative models such as the Midjourney and ChatGPT [3]. Each user denoted as u_i, is required to register before usage and would be then assigned a unique ID i where $i \in \{1, \ldots, m\}$, representing the user's unique identity. One of the users u_i would use the provided generation model to generate a picture $\mathbf{p}$. In practice, the $\mathbf{p}$ might be further processed under several image manipulations f such as resizing and compression. We aim to build a security auditing system to enable the detection and tracing of the diffusion-generated content $\mathbf{p}$.

We divided our task into two challenges. The first challenge involves determining whether the generative content originates from our diffusion model, which we refer to as detection. However, simple detection of the generative content is insufficient to differentiate who generated a particular image when multiple users are engaged. Therefore, our second challenge focuses on a more complex scenario, which entails accurate identification of the owner of $f(\mathbf{p})$, from a pool of users. To address both challenges effectively, we propose a unified watermarking framework named WaDiff, which incorporates the embedding of user-specific fingerprints within the generation process. In other words, the generative images of each user are watermarked with unique information that distinguishes them from other users and facilitates both detection and identification.

4 Methodology

4.1 Preliminaries of Diffusion Models

In this section, we briefly introduce the notation for a vanilla DDPM. The basic diffusion model typically involves two critical components known as the forward and backward processes. The forward process is a Markov chain that gradually adds noise into a real data sample $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ over T steps. Specifically, for each time step $t \in \{1, \ldots, T\}$, the latent $\mathbf{x}_t$ is obtained by adding Gaussian noise to the previous latent $\mathbf{x}_{t-1}$. Alternately, the noising procedure can be viewed as sampling from the distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$, which could be re-parameterized as:

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon, \tag{1}$$

where $\epsilon \in \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a Gaussian noise, and β_t is a predefined time-dependent variance schedule. By substituting $\alpha_t = 1 - \beta_t$ and $\overline{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, the Equation 1 could be further simplified to its closed form:

$$\mathbf{x}_t = \sqrt{\overline{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon. \tag{2}$$

Contrary to the forward process, the backward process aims to gradually reverse each forward step by estimating the latent $\mathbf{x}_{t-1}$ from $\mathbf{x}_t$. Starting from the $\mathbf{x}_T$, the diffusion model parameterized by θ, predicts the learned estimate of the Gaussian noise in Equation 2 as $\epsilon_\theta(\mathbf{x}_T)$ and then the $\mathbf{x}_{T-1}$ could be sampled from the distribution denoted as $p(\mathbf{x}_{T-1}|\mathbf{x}_T, \epsilon_\theta(\mathbf{x}_T))$. In the subsequent denoising procedure, a similar recovery from $\mathbf{x}_t$ to $\mathbf{x}_{t-1}$ is repeated until the restoration of the original input $\mathbf{x}_0$.

4.2 Rooting Fingerprints in Diffusion Process

Pre-training Watermark Decoders. Inspired by [11], we first train an image encoder $\mathcal{W}$ and decoder $\mathcal{D}$ such that $\mathcal{D}$ could retrieve the message $\mathbf{w}$ pre-embedded by $\mathcal{W}$. The $\mathcal{W}$ is then discarded and only the pre-trained $\mathcal{D}$ is left to serve as a reference to fine-tune the diffusion model. Specifically, the encoder $\mathcal{W}$ takes image $\mathbf{x}$ and n-bit binary message $\mathbf{w} \in \{0, 1\}^n$ as input and encodes $\mathbf{w}$ as an imperceptible residual δ added to $\mathbf{x}$, where the extractor $\mathcal{D}$ aims to restore the pre-encoded $\mathbf{w}$ from the watermarked input $\mathcal{W}(\mathbf{x}) = \mathbf{x} + \delta$. To train $\mathcal{W}$ and $\mathcal{D}$ in an end-to-end manner, we utilize two loss terms to fulfill the optimization objective. First, to accurately restore $\mathbf{w}$ from $\mathcal{W}(\mathbf{x})$, we minimize the Binary Cross Entropy (BCE) between the decoded output $\mathcal{D}(\mathcal{W}(\mathbf{x}))$ and ground-truth $\mathbf{w}$. Second, to reduce the visibility of δ, we penalize $\|\delta\|_2$ such that the added message perturbation is less perceptible. We further notice that the $\mathcal{D}$ obtained with the above training schedule is sensitive to several data manipulations such as resizing and compression. Therefore, similar to the simulation layer in [11], we transform $\mathcal{W}(\mathbf{x})$ with random data augmentations during the training stage to improve the robustness of the watermark system. We formulate the complete training objective as follows:

$$\min_{\mathcal{W},\mathcal{D}} \mathbb{E}_{\mathbf{x},\mathbf{w},f}[\mathcal{L}_{BCE}(\mathcal{D}(f(\mathcal{W}(\mathbf{x}))), \mathbf{w}) + \gamma\|\delta\|_2], \tag{3}$$

where $\gamma > 0$ is a hyperparameter to control the visibility of δ and f is a randomly selected image transformation from a pool of data augmentations.

Watermark-conditioned Diffusion Model. In contrast to post-hoc strategies, where the fingerprinting process occurs after the entire generation process, our approach imprints fingerprints during the sampling process. To discern the source user of generated outputs, we assign each user u_i a unique $\mathbf{w}_i$ as the conditioned input of the diffusion model such that $\mathbf{w}_i$ could be embedded to the generated content during inference and correctly restored by the pre-trained $\mathcal{D}$.

To incorporate $\mathbf{w}_i$ into the generation process, we expand the input channels and conceal the watermarking information at each denoising step t. This is achieved by applying a linear layer to project $\mathbf{w}_i$ into $\mathcal{P}(\mathbf{w}_i) \in \mathbb{R}^{\widetilde{C}\times H\times W}$, where $\widetilde{C}$ denotes the number of watermark channels in the latent space. We then concatenate it with the original latent variable $\mathbf{x}_t \in \mathbb{R}^{C\times H\times W}$ along their first dimension, resulting in the conditioned latent variable $\hat{\mathbf{x}}_{t,i} = concat(\mathbf{x}_t, \mathcal{P}(\mathbf{w}_i)) \in \mathbb{R}^{(C+\widetilde{C})\times H\times W}$. Note that channel expansion is a common manipulation to integrate additional information and has been widely used in previous works [32,45]. The next step involves embedding $\mathbf{w}_i$ into the generative content to allow for detection by the pre-trained decoder $\mathcal{D}$. However, as the sampling process typically involves multiple denoising steps to obtain the generative output, it is challenging to directly incorporate it into the fine-tuning process. To address this issue, we start by restoring the original image $\mathbf{x}_0$ within a single step. Revisiting the forward noising in Equation 2, $\mathbf{x}_0$ could be directly recovered from $\mathbf{x}_t$ by subtracting the second noise term and subsequently scaling. In the absence of the ground truth ϵ, we take the prediction from the diffusion model as an estimate. In our watermark-conditioned model, we replace ϵ with $\epsilon_\theta(\hat{\mathbf{x}}_{t,i})$ and construct the conditioned reverse of $\mathbf{x}_0$ at time step t as:

$$\mathbf{x}_{0,i}^t = \frac{\mathbf{x}_t - \sqrt{1-\overline{\alpha}_t}\epsilon_\theta(\hat{\mathbf{x}}_{t,i})}{\sqrt{\overline{\alpha}_t}}. \tag{4}$$

We then formulate our first optimization objective as optimizing the **message retrieval loss**, which we defined as follows:

$$\min_\theta \mathbb{E}_{\mathbf{x},i,t}[\mathcal{L}_m(\mathcal{D}(\mathbf{x}_{0,i}^t), \mathbf{w}_i)]. \tag{5}$$

By optimizing Equation 5, we ensure that the pre-embedded $\mathbf{w}_i$ in $\mathbf{x}_{0,i}^t$ could be detected by $\mathcal{D}$. However, the image quality of $\mathbf{x}_{0,i}^t$ is significantly influenced by the denoising step t, resulting in higher image quality for smaller t and noisier images for larger t. Therefore, we empirically introduce a threshold τ where we only minimize the message retrieval loss when $t \leq \tau$. This strategy helps effectively inject the watermark while stabilizing the fine-tuning procedure.

Preserving Image Consistency. In addition to fingerprinting the output, it is critical to preserve the generated images' quality after watermarking. In other words, the generated images after watermarking across different users should be visually equivalent. To achieve this, we treat the original model as an oracle and align the generated output with the original output. Since the whole generation involves multiple denoising steps, we need to ensure the $\epsilon_\theta(\hat{\mathbf{x}}_{t,i})$ and $\epsilon_{\theta_{ori}}(\mathbf{x}_t)$ are aligned for each t, where θ_{ori} represents the pre-trained diffusion model weights. Therefore, we introduce the second optimization objective by minimizing over the image **consistency loss**, denoted as follows:

$$\min_{\theta} \mathbb{E}_{\mathbf{x},i,t}[\mathcal{L}_c(\epsilon_\theta(\hat{\mathbf{x}}_{t,i}), \epsilon_{\theta_{ori}}(\mathbf{x}_t))], \tag{6}$$

where $\mathcal{L}_c$ is the Mean-Squared Loss. To further improve the image consistency between different watermarks, for $t > \tau$, we replace the conditioned input $\mathbf{w}_i$ with a never-used null watermark denoted as $\mathbf{w}_{null}$. By fixing the condition with $\mathbf{w}_{null}$, we ensure that $\epsilon_\theta(\hat{\mathbf{x}}_{t,null})$ is distinct from $\mathbf{w}_i$ and keeps unchanged until $t \leq \tau$, which yields an improved image consistency among diverse users. We demonstrate watermarked samples of our method along with Tree-Ring [34] in Figure 2 and can observe that our method significantly improves the image consistency between watermarked contents. From the results, our watermarked outputs are not only visually equivalent across different users but also maintain the original semantic meaning with only a slight visual difference.

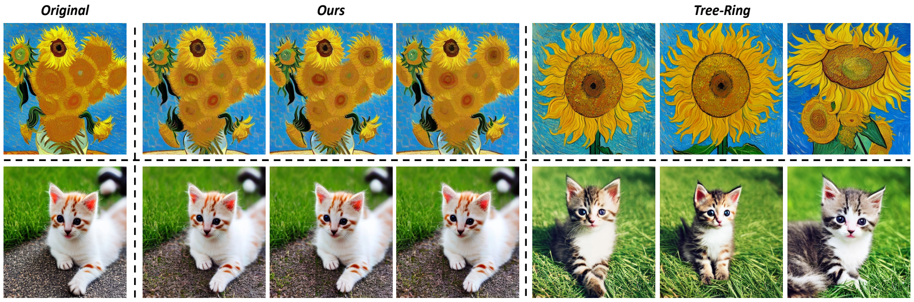

Fig. 2: Watermarked examples of our method and Tree-Ring$_{Rings}$ sampled from the Stable Diffusion. It is observed that our method achieves a substantial improvement in image consistency among images with diverse watermarks.

End-to-end Fine-tuning. Instead of fine-tuning the entire diffusion model, we selectively fine-tune the watermark projector $\mathcal{P}$ and the first input block while keeping the remaining weights unchanged. We adopt this approach for two reasons. First, we observe that fine-tuning the first block is sufficient for effective watermark injection. Our method achieves comparable performance to fine-tuning the entire architecture but with faster speed and lower memory cost. Additionally, we have empirically observed that fine-tuning the entire model can lead to unstable generation, resulting in watermarked images with significantly compromised generative quality. A detailed analysis of fine-tuning various model subsections is provided in the *Appendix*. Formally, we separate the whole parameters θ into two sets: θ_{head}, representing the parameters of $\mathcal{P}$ and the first input block, and θ_{tail}, representing the remaining weights, *i.e.*, $\theta = \{\theta_{head}, \theta_{tail}\}$. During fine-tuning, only θ_{head} is optimized while θ_{tail} is fixed. In sum, we incorporate both optimization objectives above and formulate the fine-tuning process as:

$$\begin{aligned}\min_{\theta_{head}} \mathbb{E}_{\mathbf{x},i,t}[&\mathbb{I}(t \le \tau)(\mathcal{L}_c(\epsilon_\theta(\hat{\mathbf{x}}_{t,i}), \epsilon_{\theta_{ori}}(\mathbf{x}_t)) + \eta\mathcal{L}_m(\mathcal{D}(\mathbf{x}^t_{0,i}), \mathbf{w}_i)) \\ &+\mathbb{I}(t > \tau)\mathcal{L}_c(\epsilon_\theta(\hat{\mathbf{x}}_{t,null}), \epsilon_{\theta_{ori}}(\mathbf{x}_t))],\end{aligned} \tag{7}$$

where $\mathbb{I}$ represents the indicator function and η controls the trade-off between image consistency and watermarking effectiveness. Our framework is demonstrated in Figure 3.

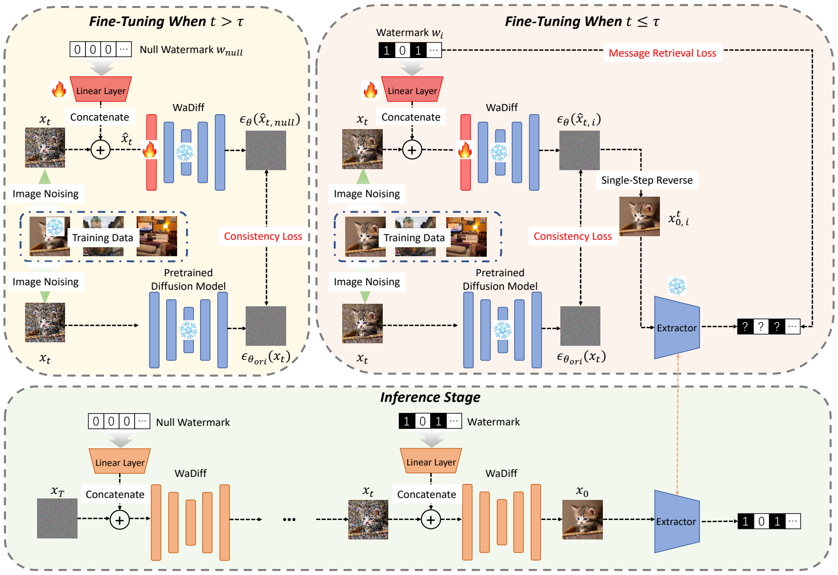

Fig. 3: Overview of our proposed watermarking framework. The top two figures illustrate our fine-tuning process. For $t > \tau$, we solely focus on preserving image consistency and incorporate a null watermark. For $t \le \tau$, we integrate the normal watermark and introduce an additional message retrieval loss to embed watermarks. The inference stage is depicted below, where we inject the null watermark when $t > \tau$ and transition it to the payload watermark when $t \le \tau$.

4.3 Detection and Identification

Once the diffusion model is fingerprinted, the generated content for each u_i will contain specific information that is conditioned by $\mathbf{w}_i$. Given a candidate $\mathbf{p}$, we can then recover the source watermark $\mathbf{w}_s = \mathcal{D}(\mathbf{p})$ using the pre-trained $\mathcal{D}$ model. Once we have recovered $\mathbf{w}_s$, we can utilize it for both detection and identification. To detect whether $\mathbf{p}$ belongs to our model, we set a bit threshold τ_b and calculate the number of matched bits $M_i = \mathbf{w}_s \odot \mathbf{w}_i$, where $\odot$ represents the XNOR function. If $\frac{M_i}{n} > \tau_b$, we will conclude that $\mathbf{p}$ is generated from our model; otherwise, it is generated from other sources. For identification among m users, we can determine the owner by finding the corresponding user ID whose watermark $\mathbf{w}_i$ has the best match with $\mathbf{w}_s$, which can be formulated as follows:

$$\arg\max_i M_i, \quad i \in \{1 \dots m\}. \tag{8}$$

5 Experiments

To evaluate the identification performance of our proposed WaDiff, we implement our method on two widely used diffusion models: a text-to-image latent diffusion model and an unconditional diffusion model. We conduct comprehensive comparisons with other watermarking strategies including a traditional strategy DwtDct [6], a deep-learning-based steganography technique StegaStamp [32], which we employed in our pre-training stage, and a recently proposed training-free framework Tree-Ring [34]. Note that we have not included the Stable Signature [11] in our comparison since it considered the model distribution setting which is different from ours. Besides, it necessitates re-training diverse latent decoders, which is not scalable in our identification task. We also conduct thorough ablation studies and provide detailed analysis in the following sections.

5.1 Experimental Settings

Model and Dataset. For the text-to-image latent diffusion model, we utilize the Stable Diffusion V1.4 as our pre-trained model, and for the unconditional diffusion model, we utilize the 256 × 256 ImageNet diffusion model. To fine-tune the diffusion models, we randomly select 5000 images with corresponding text descriptions from the training set of MS-COCO 2014 [18] for the Stable Diffusion and 5000 images from the training set of ImageNet [8] for the 256x256 ImageNet diffusion model.

To benchmark the effectiveness of our proposed strategy, we employ different metrics for detection and identification evaluation. For detection, we report the area under the curve (AUC) of the receiver operating characteristic (ROC) curve. For identification, we begin by generating a pool of m users, each associated with a unique binary code. Randomly drawn from this pool, we select a subset of users and generate images for each selected user. The identification performance of our method is evaluated using the tracing accuracy metric defined as:

$$\text{Trace Acc} = \frac{N_{\text{correct}}}{N_{\text{total}}}, \tag{9}$$

where N_{correct} represents the number of correctly identified images and N_{total} is the total number of candidate images for identification. Additionally, we assess the image consistency by calculating the structural similarity index (SSIM) [33] between pairs of watermarked content. Furthermore, we measure the impact of watermarking on the original generation quality by reporting the difference between the Frechet Inception Distance (FID) [13] of the watermarked contents and the originally generated contents.

Implementation Details. All experiments were conducted utilizing 8 NVIDIA A100 GPUs. The fine-tuning of both diffusion models involved employing an AdamW optimizer [20] with a learning rate of $1e^{-4}$. For the Stable Diffusion model, we set τ to 500 and η to 0.05, while for the ImageNet diffusion model, we selected τ as 400 and η as 0.25. Notably, in addition to directly aligning the predicted noise of the ImageNet diffusion model with the original model, we empirically found that aligning the single-step-reverse images resulted in improved image quality. Implementation details and additional hyperparameter evaluations are deferred in the *Appendix*. We fine-tune 40 epochs for the Stable Diffusion model and 25 epochs for the ImageNet diffusion model. For the Stable Diffusion model, we set the default guidance scale to 7.5 and used text descriptions from the validation set of MS-COCO 2014 [18] as prompts. We adopt the watermark length to 48 by default, which is commonly used in previous work [11], and provide experiments of other lengths in the *Appendix*. We adopted the DDIM [30] sampler with 50 sampling steps as default for both models. More implementation details on training baseline methods and the watermark decoder pre-training is in the *Appendix*.

Table 1: This table includes our main results. Trace m indicates the tracing accuracy (%) of our identification among m users in total.

Model	Type	Method	AUC	Trace 10^4	Trace 10^5	Trace 10^6	Trace Avg	SSIM(↑)	FID Diff(↓)
Stable Diffusion	Post Generation	DwtDct	0.917	76.30	74.70	72.90	74.63	0.999	-0.36
		StegaStamp	1.000	99.98	99.98	99.96	99.97	0.999	+0.27
	Merged Generation	Tree-Ring$_{Rand}$	0.999	0.04	0.00	0.00	0.01	0.457	+0.14
		Tree-Ring$_{Rings}$	0.999	0.00	0.00	0.00	0.00	0.575	+0.77
		WaDiff (Ours)	0.999	98.20	96.76	93.44	96.13	0.999	+0.41
256×256 ImageNet	Post Generation	DwtDct	0.936	71.30	68.10	65.20	68.20	0.997	-0.05
		StegaStamp	1.000	99.98	99.98	99.98	99.98	0.998	+0.11
	Merged Generation	Tree-Ring$_{Rand}$	0.999	0.00	0.00	0.00	0.00	0.584	+0.17
		Tree-Ring$_{Rings}$	0.999	0.00	0.00	0.00	0.00	0.652	+0.23
		WaDiff (Ours)	1.000	99.68	99.38	98.78	99.28	0.997	+0.08

5.2 Detection and Identification Results

In this section, we present the main experimental results of our method. For the detection task, we use 5000 watermarked images along with another 5000 clean images sampled from the original pre-trained diffusion model to calculate the AUC. As for owner identification, we evaluate our method using different sizes of user pools, ranging from ten thousand to one million users. For each user pool, we randomly select 1000 users and generate 5 images per user, resulting in a total of 5000 images. The tracing accuracy is then calculated based on these watermarked images. For SSIM calculation, we first randomly select 200 distinct initial noises and generate a group of 5 images for each noisy vector with different keys. Within each group, the SSIM metric was computed by comparing the similarity between each pair of images. To calculate the FID difference, we evaluate the generative images (for both watermarked and originally generative contents) on the MS-COCO 2014 training set for Stable Diffusion and on the ImageNet [8] training set for the ImageNet diffusion model. We present the experimental results in Table 1. The results demonstrate that our method achieves comparable performance in both detection and identification tasks when compared to the post-hoc StegaStamp, achieving a tracing accuracy of 97.71% and an AUC of 1 on average. We defer further discussion on the comparison with post-hoc methods to the *Appendix*. In terms of image consistency, our method achieves an average of 0.998 SSIM between pairs of images with different watermarks, which significantly surpasses that of Tree-Ring. Besides, our method only imposes a negligible impact on the original generation quality by a slight increase on the original FID, which is comparable to the post-hoc watermark. Note that the Tree-Ring achieves nearly zero performance on the identification task. This might be attributed to the **imprecise latent inversion** and the **continuous watermarking space**, which makes watermarked examples less distinguishable when multiple users are engaged.

Table 2: This table reports WaDiff tracing accuracy (%) and AUC under diverse data augmentations.

Model	Case	Resize	Blurring	Color Jitter	Noising	JPEG	Combine	Avg
Stable Diffusion	AUC	0.999	0.999	0.999	0.997	0.999	0.999	0.999
	Trace 10^4	97.02	97.14	96.00	88.52	93.48	93.02	94.19
	Trace 10^5	94.34	94.12	88.56	81.14	87.66	84.26	88.34
	Trace 10^6	89.46	87.40	82.14	72.50	80.30	78.04	81.64
256×256 ImageNet	AUC	0.999	0.999	0.999	0.999	0.999	0.999	0.999
	Trace 10^4	98.90	94.48	98.56	91.80	92.06	91.88	94.61
	Trace 10^5	97.78	89.90	96.48	84.46	88.70	85.74	90.51
	Trace 10^6	96.02	82.42	94.50	76.26	77.88	76.88	83.99

5.3 Watermark Robustness Analysis

To evaluate our watermarking strategy against potential data augmentations, we adopt five commonly used augmentations including image resizing, image blurring, color jitter, Gaussian noising, and JPEG compression. Specifically, for resizing, we randomly resize the width and height of images within a range of 30% to 80% of their original size; for image blurring, we adopt the kernel size to 20; for color jitter, we randomly select from saturation (factor 1.5), contrast (factor 1.5) and sharpness (factor 1.5); for Gaussian noising, we add a Gaussian noise with $\sigma = 0.1$; for the JPEG compression, we select the compression quality as 50. We also consider a combinational augmentation that incorporates image resizing, JPEG compression, and color jitter simultaneously. We report the identification performance under various augmentations in Table 2. The experimental results demonstrate the overall robustness of our method against diverse image augmentations, yielding an average tracing accuracy of 88.05% for the

Stable Diffusion model and 89.7% for the ImageNet diffusion model. It is noteworthy that our method maintains an average AUC of 0.999 under various data augmentations and achieves identification performance of over 76% among one million users even under combined data augmentation, which further validates the practicality and effectiveness of WaDiff. We provide robustness comparisons with other watermarking schemes in the *Appendix*.

5.4 Ablation Study

Detection with Different Bit Thresholds. In addition to the reported metric AUC in our main results, in this section, we make a detailed analysis of our detection capability. In the detection task, the choice of the bit threshold τ_b would impact the overall detection performance. To investigate this, we vary the value of τ_b from 0.65 to 0.85 and examine the precision and recall of our detection performance. We conduct experiments on a dataset consisting of 5000 clean images and 5000 watermarked images, and the results are presented in Table 3. Overall, our method demonstrates robustness to the selection of τ_b, indicating its stability and reliability in detecting generative content.

Table 3: Detection results against different bit thresholds. P_S and R_S indicate the precision and recall for the Stable Diffusion respectively, where P_I and R_I represent the precision and recall for the ImageNet diffusion model respectively.

Metric	$\tau_b = 0.65$	$\tau_b = 0.7$	$\tau_b = 0.75$	$\tau_b = 0.8$	$\tau_b = 0.85$	AVG
P_S	0.982	0.997	1.000	1.000	1.000	0.996
R_S	0.998	0.994	0.978	0.951	0.898	0.964
P_I	0.989	0.998	0.999	1.000	1.000	0.997
R_I	0.999	0.999	0.997	0.994	0.983	0.994

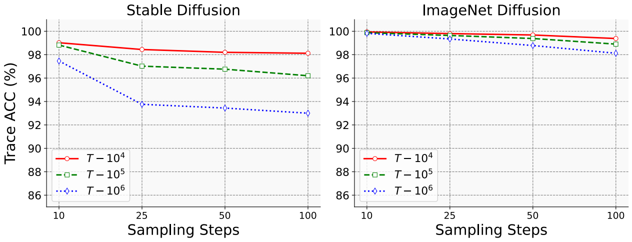

Fig. 4: The tracing accuracy results of two diffusion models with different DDIM sampling steps. We denote $T-m$ as tracing among m users.

Experiments on Different Sampling Steps. In this section, we analyze the impact of different sampling steps on our watermarking results. In addition to the 50 steps as default, we verify our method on 10, 25, and 100 DDIM sampling steps and demonstrate our experimental results in Figure 4. The results demonstrate that our WaDiff achieves a stable performance across various sampling steps. Notably, it is observed that with a smaller sampling step, the tracing accuracy is boosted for both models, especially for the Stable Diffusion model, which shows an average increase of 2.3% tracing accuracy for 10 sampling steps compared to the default 50 steps. This can be attributed to the fact that when the sampling step is small, the noisy image is not effectively recovered, allowing our watermark information to be more easily concealed in these flaw areas.

5.5 Robustness Analysis with Adaptive Attacks

In addition to the commonly used image augmentations discussed in Section 5.3, we investigate two additional countermeasures to assess the robustness of our watermarking scheme. For the **Diff** Attack, we adopt a method similar to DiffPure [24], which uses a pre-trained diffusion model to purify our watermarked images. In our experiments, we set the number of diffusion and denoising steps to 20 and utilize the original DDPM sampler for this attack. In the **Multi-Message** Attack, we train a surrogate watermark encoder with the same architecture as in our experiments, but using different training data. We then embed a random watermark message into the watermarked example. Evaluation results against two adaptive attacks on both WaDiff and StegaStamp are presented in Table 4. Our results demonstrate that WaDiff exhibits superior robustness compared to StegaStamp in these two adaptive attacks. This can be attributed to the integrated design of our watermarks, which are less vulnerable to diffusion denoising and are not easily removed by directly injecting post-hoc information.

Table 4: The tracing accuracy against two adaptive attacks, where DA and MA are short for the Diff and Multi-Message attack respectively.

Attack	Trace 10^4	Trace 10^5	Trace 10^6
DA (WaDiff)	77.50%	63.34%	49.42%
DA (Stega)	0.10%	0.00%	0.00%
MA (WaDiff)	97.96%	95.66%	91.48%
MA (Stega)	0.02%	0.00%	0.00%

6 Conclusion and Limitations

In this paper, we provide an efficient and robust watermarking framework WaDiff to not only detect whether an image is generated from our model but also identify the specific user who generated the image. WaDiff seamlessly incorporates a user-specific watermark as a conditioned input and applies fingerprinting to the generated contents during the image generation process. Extensive experiments demonstrate that WaDiff achieves accurate identification performance among a large number of users and remains robust under various data augmentations. While WaDiff still exhibits slightly inferior performance compared to the post-hoc watermarking method, our method fingerprints in the generation process which is more stealthy and hard to circumvent in practice. We hope that our work can pioneer the secure auditing of AI-generated content so that we can ensure that model usage aligns with regulatory requirements, compliance, and ethical standards, thereby enhancing the integrity of the generative models.

References

1. Barni, M., Bartolini, F., Cappellini, V., Piva, A.: A dct-domain system for robust image watermarking. Signal processing **66**(3), 357–372 (1998)
2. Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., Manassra, W., Dhariwal, P., Chu, C., Jiao, Y., Ramesh, A.: Improving image generation with better captions (2023)
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
4. Cheng, M., Min, R., Sun, H., Chen, P.Y.: Identification of the adversary from a single adversarial example. In: International Conference on Machine Learning. pp. 5472–5484. PMLR (2023)
5. Ci, H., Yang, P., Song, Y., Shou, M.Z.: Ringid: Rethinking tree-ring watermarking for enhanced multi-key identification. arXiv preprint arXiv:2404.14055 (2024)
6. Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: Digital watermarking and steganography. Morgan kaufmann (2007)
7. Cui, Y., Ren, J., Xu, H., He, P., Liu, H., Sun, L., Tang, J.: Diffusionshield: A watermark for copyright protection against generative diffusion models. arXiv preprint arXiv:2306.04642 (2023)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
9. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)
10. Fernandez, P., Chaffin, A., Tit, K., Chappelier, V., Furon, T.: Three bricks to consolidate watermarks for large language models. arXiv preprint arXiv:2308.00113 (2023)
11. Fernandez, P., Couairon, G., Jégou, H., Douze, M., Furon, T.: The stable signature: Rooting watermarks in latent diffusion models. arXiv preprint arXiv:2303.15435 (2023)
12. Ganic, E., Eskicioglu, A.M.: Robust dwt-svd domain image watermarking: embedding data in all frequencies. In: Proceedings of the 2004 Workshop on Multimedia and Security. pp. 166–174 (2004)
13. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
14. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
15. Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., Goldstein, T.: A watermark for large language models. arXiv preprint arXiv:2301.10226 (2023)
16. Li, Y., Bai, Y., Jiang, Y., Yang, Y., Xia, S.T., Li, B.: Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection. Advances in Neural Information Processing Systems **35**, 13238–13250 (2022)
17. Li, Y., Li, Y., Wu, B., Li, L., He, R., Lyu, S.: Invisible backdoor attack with sample-specific triggers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 16463–16472 (2021)
18. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)

19. Liu, Y., Guo, M., Zhang, J., Zhu, Y., Xie, X.: A novel two-stage separable deep learning framework for practical blind watermarking. In: Proceedings of the 27th ACM International conference on multimedia. pp. 1509–1517 (2019)
20. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
21. Ma, Y., Zhao, Z., He, X., Li, Z., Backes, M., Zhang, Y.: Generative watermarking against unauthorized subject-driven image synthesis. arXiv preprint arXiv:2306.07754 (2023)
22. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. arXiv preprint arXiv:2108.01073 (2021)
23. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
24. Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., Anandkumar, A.: Diffusion models for adversarial purification. arXiv preprint arXiv:2205.07460 (2022)
25. Nightingale, S.J., Farid, H.: Ai-synthesized faces are indistinguishable from real faces and more trustworthy. Proceedings of the National Academy of Sciences **119**(8), e2120481119 (2022)
26. Nikolaidis, N., Pitas, I.: Robust image watermarking in the spatial domain. Signal processing **66**(3), 385–403 (1998)
27. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
28. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in Neural Information Processing Systems **35**, 36479–36494 (2022)
29. Singh, A.K., Sharma, N., Dave, M., Mohan, A.: A novel technique for digital image watermarking in spatial domain. In: 2012 2nd IEEE International Conference on Parallel, Distributed and Grid Computing. pp. 497–501. IEEE (2012)
30. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
31. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
32. Tancik, M., Mildenhall, B., Ng, R.: Stegastamp: Invisible hyperlinks in physical photographs. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2117–2126 (2020)
33. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
34. Wen, Y., Kirchenbauer, J., Geiping, J., Goldstein, T.: Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. arXiv preprint arXiv:2305.20030 (2023)
35. Wu, H., Liu, G., Yao, Y., Zhang, X.: Watermarking neural networks with watermarked images. IEEE Transactions on Circuits and Systems for Video Technology **31**(7), 2591–2601 (2020)
36. Xiong, C., Qin, C., Feng, G., Zhang, X.: Flexible and secure watermarking for latent diffusion model. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 1668–1676 (2023)

37. Yang, Z., Zeng, K., Chen, K., Fang, H., Zhang, W., Yu, N.: 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 (2024)
38. Yu, N., Skripniuk, V., Abdelnabi, S., Fritz, M.: Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In: Proceedings of the IEEE/CVF International conference on computer vision. pp. 14448–14457 (2021)
39. Yu, N., Skripniuk, V., Chen, D., Davis, L., Fritz, M.: Responsible disclosure of generative models using scalable fingerprinting. arXiv preprint arXiv:2012.08726 (2020)
40. Zhang, C., Benz, P., Karjauv, A., Sun, G., Kweon, I.S.: Udh: Universal deep hiding for steganography, watermarking, and light field messaging. Advances in Neural Information Processing Systems **33**, 10223–10234 (2020)
41. Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M.P., Huang, H., Molloy, I.: Protecting intellectual property of deep neural networks with watermarking. In: Proceedings of the 2018 on Asia conference on computer and communications security. pp. 159–172 (2018)
42. Zhang, J., Chen, D., Liao, J., Fang, H., Zhang, W., Zhou, W., Cui, H., Yu, N.: Model watermarking for image processing networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 12805–12812 (2020)
43. Zhao, Y., Pang, T., Du, C., Yang, X., Cheung, N.M., Lin, M.: A recipe for watermarking diffusion models. arXiv preprint arXiv:2303.10137 (2023)
44. Zhong, X., Huang, P.C., Mastorakis, S., Shih, F.Y.: An automated and robust image watermarking scheme based on deep neural networks. IEEE Transactions on Multimedia **23**, 1951–1961 (2020)
45. Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: Hidden: Hiding data with deep networks. In: Proceedings of the European conference on computer vision (ECCV). pp. 657–672 (2018)