RAW: A Robust and Agile Plug-and-Play Watermark Framework for AI-Generated Images with Provable Guarantees

Xun Xian [1] **Ganghua Wang** [2] **Xuan Bi** [3] **Jayanth Srinivasa** [4] **Ashish Kundu** [4] **Mingyi Hong** [1] **Jie Ding** [2]

Abstract

Safeguarding intellectual property and preventing potential misuse of AI-generated images are of paramount importance. This paper introduces a robust and agile plug-and-play watermark detection framework, referred to as RAW. As a departure from existing encoder-decoder methods, which incorporate fixed binary codes as watermarks within latent representations, our approach introduces learnable watermarks directly into the original image data. Subsequently, we employ a classifier that is jointly trained with the watermark to detect the presence of the watermark. The proposed framework is compatible with various generative architectures and supports on-the-fly watermark injection after training. By incorporating state-of-the-art smoothing techniques, we show that the framework also provides provable guarantees regarding the false positive rate for misclassifying a watermarked image, even in the presence of adversarial attacks targeting watermark removal. Experiments on a diverse range of images generated by state-of-the-art diffusion models demonstrate substantially improved watermark detection performance and/or watermark encoding speed, under adversarial attacks, while maintaining image quality.

1. Introduction

In recent years, Generative Artificial Intelligence, notably in computer vision, has made significant strides. The adoption of diffusion models (DM) in applications like Stable Diffusion (Rombach et al., 2022) and DALLE-2 (Ramesh et al., 2022) has greatly improved image generation quality. However, these advancements also raise concerns about potential misuse, seen in instances such as DeepFake (Verdoliva, 2020) and copyright infringement (Sag, 2023).

To mitigate the potential misuse of diffusion models, the incorporation of watermarks emerges as a promising solution. Watermarked images, tagged with crafted signals, act as markers to identify their machine-generated origin. Watermarking techniques designed for generative models can be generally classified into two categories: model-specific (Fernandez et al., 2023; Nie et al., 2023; Kim et al., 2023; Wen et al., 2023) and model-agnostic (Cox et al., 1996; Zhang et al., 2019; Tancik et al., 2020), as outlined in Table 1. Model-specific methods refer to those tailored for particular generative models, often offering a better tradeoff between watermarked image quality and watermark detection performance. However, this characteristic may potentially restrict their applicability across diverse use cases. For instance, the Tree-Ring watermark (Wen et al., 2023) is tailored for specific samplers, e.g., DDIM (Song et al., 2020), used for image generation within diffusion models. The feasibility of adapting this method to other commonly used samplers remains open to discussion.

In contrast, model-agnostic approaches directly watermark generated content without modifying the generative models. These approaches can be categorized into two types. The first, from traditional signal processing, e.g., DwTDcT (Cox et al., 2007), embeds watermarks in specific parts of images' frequency domains. However, they are vulnerable to strong image manipulations and adversarial attacks for removing watermarks (Ballé et al., 2018). The second type employs deep learning techniques, utilizing encoder-decoder structures to embed watermarks, such as binary codes, in latent spaces. For instance, RivaGan (Zhang et al., 2019) trains the watermark and watermark decoder jointly as learned models, enhancing transmission and robustness. However, these methods demand greater computational resources for watermark injection, limiting real-time on-the-fly deployment. For example, when injecting watermarks into images, the RivaGan method requires over $15\times$ the time needed by the DwtDct methods (Wang, 2022).

Additionally, there has been a sustained emphasis on precisely evaluating false-positive rates (FPRs) and/or the Area Under the Receiver Operating Characteristic curve (AUROC) for each utilized watermarking strategy (Pitas, 1998),

[1]Department of ECE, University of Minnesota [2]School of Statistics, University of Minnesota [3]Carlson School of Management, University of Minnesota [4]Cisco Research. Correspondence to: Jie Ding <dingj@umn.edu>.

Under Review

Table 1: Summary of features of several representative watermark techniques. The second column denotes the method's suitability for real-time on-the-fly implementation. / denotes cases where the watermark is embedded during the generative process. The third column evaluates whether the watermarking method provides provable guarantees on false-positive rates (FPRs) under adversarial attacks in a distribution-free manner.

Method	*Model agnostic*	*On-the-fly deployment*	*Provable FPRs under adversarial attacks*
DwtDct (Cox et al., 2007)	✓	✓	✗
RivaGan (Zhang et al., 2019)	✓	✗	✗
StegStamp (Tancik et al., 2020)	✓	✗	✗
Stable Signature (Fernandez et al., 2023)	✗	/	✗
Tree Ring (Wen et al., 2023)	✗	/	✗
RAW (Our method)	✓	✓	✓

given the potential economic implications associated with watermark implementation. To establish an explicit theoretical formulation for FPRs, many studies have assumed that the binary watermark code extracted from unwatermarked images exhibits a pattern where each digit is an independent and identically distributed (IID) Bernoulli random variable with equal probabilities. This assumption enables the explicit derivation of the FPRs when comparing the extracted binary code with the predefined actual binary watermark code. However, such an assumption may not hold as empirically observed in (Fernandez et al., 2022), and thus the corresponding formulation for FPRs could be incorrect.

1.1. Contributions

In this paper, we introduce a **R**obust, **A**gile plug-and-play **W**atermark framework, abbreviated as **RAW**. RAW is designed for both adaptability and computation efficiency, providing a model-agnostic approach for real-time, on-the-fly deployment of image watermarking. This dedication to adaptability is to ensure the accessibility for third-party users, e.g., artists and generative model providers. Moreover, this adaptability is fortified by the integration of state-of-the-art smoothing techniques for achieving provable guarantees on the FPRs for detection, even under adversarial attacks.

A new framework for robust and agile plug-and-play watermark learning. As illustrated in Figure 1, in contrast to encoder-decoder techniques that insert fixed binary watermarks into latent spaces, we propose to embed learnable watermarks, matching the image dimensions, into both the frequency and spatial domains of the original images. To differentiate between watermarked and unwatermarked samples, we utilize a classifier, e.g., a convolutional neural network (CNN), and perform joint training for both the watermarks and the classifier. The proposed framework offers several benefits, including enhanced computational efficiency achieved through batch processing for watermark injection post joint training. For instance, our experimental results demonstrate time efficiency enhancements, approximately $30\times$ ($200\times$) faster than the frequency-based

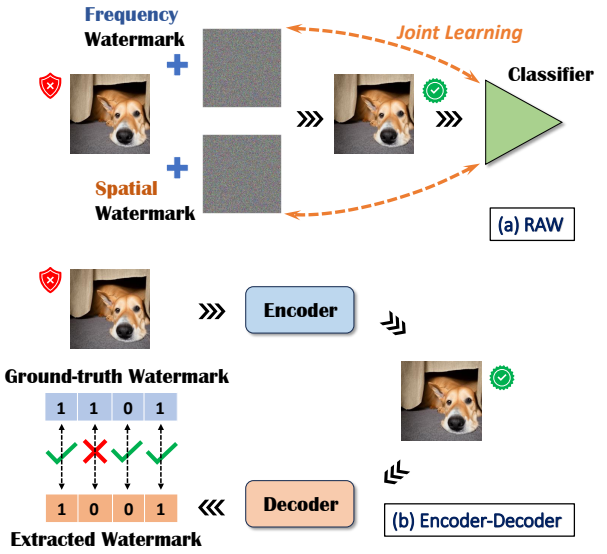

Figure 1: Illustration of our proposed RAW (top row) and popular encoder-decoder based watermarking schemes (bottom row). In RAW, watermarks are directly embedded into images, eliminating the need for a complex encoder, which enhances suitability for on-the-fly deployment.

(encoder-decoder based) method, respectively. Moreover, it can be readily integrated with other state-of-the-art techniques to further enhance robustness and generalizability, such as adversarial training (Goodfellow et al., 2014; Madry et al., 2017), contrastive learning (Chen et al., 2020; Khosla et al., 2020), and label smoothing (Müller et al., 2019).

Provable guarantees on FPRs even under adversarial attacks without distributional assumptions. By integrating advanced methods from the conformal prediction literature (Vovk et al., 2005; Lei & Wasserman, 2014) into our RAW framework, we showcase its ability to offer rigorous, *distribution-free* assurances regarding the FPRs. Additionally, we develop a novel technique, inspired by the randomized smoothing (Duchi et al., 2012; Cohen et al., 2019), to further enhance our provable guarantees. This technique ensures *certified* guarantees on FPRs under arbitrary, includ-

ing adversarial, perturbations with bounded norms. That is, as long as any transformations or adversarial attacks on future test images stay within a predefined range, our FPRs guarantees remain valid.

Extensive empirical studies on various datasets. We evaluate the efficacy of our proposed method across various generative data scenarios, such as the DBDiffusion (Wang et al., 2022b) and the generated MS-COCO (Lin et al., 2014). Our assessment includes detection performance, robustness against image manipulations/attacks, the computational efficiency of watermark injection, and the quality of generated images. The experimental results consistently affirm the excellent performance of our approach, as evidenced by notable enhancements in AUROC from 0.48 to 0.82 under state-of-the-art diffusion-model-based adversarial attacks aimed at removing watermarks.

1.2. Related Work

Classical watermarking techniques for images. Image watermarking has long been a fundamental problem in both signal processing and computer vision literature (Cox et al., 1996; Pereira et al., 1999). Methods for image-based watermarking typically operate within either the spatial or frequency domains (Cox et al., 1996; Gupta et al., 2016). Within the spatial domain, methodologies span from basic approaches, such as the manipulation of the least significant bit of individual pixels, to more complex strategies like Spread Spectrum Modulation (Altun et al., 2009; Hartung & Girod, 1998). In the frequency domain, watermark embedding (Cox et al., 1996; O'Ruanaidh & Pun, 1997) involves modifying coefficients generated by transformations such as the Discrete Cosine Transform (Hernandez et al., 2000).

Image watermarking using deep learning. In recent times, the advent of advanced deep learning techniques has opened up new avenues for watermarking. Many of these methods (Kandi et al., 2017; Hayes & Danezis, 2017; Zhu et al., 2018; Zhang et al., 2019; Fernandez et al., 2022; Tancik et al., 2020), are based on the encoder-decoder architecture. In this model, the encoder embeds a binary code into images in latent representations, while the decoder takes an image as input and generates a binary code for comparison with the binary code injected for watermark verification. For example, the HiDDeN technique (Zhu et al., 2018) involves the simultaneous training of encoder and decoder networks, incorporating noise layers specifically crafted to simulate image perturbations. While these methods enhance robustness compared to traditional watermarking, they may not be ideal for real-time, on-the-fly watermark injection due to the time-consuming feed-forward process in the encoder, particularly with larger architectures.

Watermarks for protecting model intellectual property. Deep neural networks represent valuable intellectual assets, given the significant resources invested in their training and data collection processes (Rombach et al., 2022). For example, training stable diffusion models consumes approximately 150,000 GPU hours, amounting to roughly $600,000 in costs (Wikipedia contributors, 2023). Given their broad applications in real-world scenarios, ensuring copyright protection and facilitating their identification is crucial in both normal and adversarial contexts (Tramèr et al., 2016; Xian et al., 2022). Some approaches embed watermarks directly into model parameters (Uchida et al., 2017), necessitating white-box access for inspection. Others (Adi et al., 2018; Zhao et al., 2023) rely on backdoor attacks (Gu et al., 2017; Xian et al., 2023b;a; Wang et al., 2024), injecting triggers into training data to enable targeted predictions during testing. These methods primarily focus on safeguarding model intellectual property rather than the generated outputs.

2. Preliminary

Notations. We consider the problem of embedding watermarks into images and then detecting the watermarks as a binary classification problem. Let $\mathcal{X} = [0,1]^{C\times W\times H}$ be the input space, with C, W, and H being the channel, width and height of images, respectively. We denote $\mathcal{Y} = \{0,1\}$ to be the label space, with label 0 indicating unwatermarked and 1 indicating watermarked versions, respectively. For a vector v, we use $\|v\|$ to denote its ℓ_2-norm.

Threat Model. We consider the following use scenario of watermarks between a third-party user Alice, e.g., an artist, a generative model provider Bob, e.g., DALLE-2 from OPENAI, and an adversary Cathy.

- Alice selects a diffusion model (DM) from Bob's API interface and sends an input (e.g., a prompt for text-to-image diffusion models) to Bob for generating images;
- Upon receiving Bob generated images $X \in \mathcal{X}$, Alice embeds a watermark into X, denoted as $X' \in \mathcal{X}$, and releases it to the public.
- Adversary Cathy applies (adversarial) image transformation(s), e.g., rotating and cropping, on X' to obtain a modified version $\widetilde{X}'(\in \mathcal{X})$;
- Alice decides if $\widetilde{X}'$ was generated by herself or not.

Problem Formulation. From the above, the watermark problem for Alice essentially boils down to a binary classification or hypothesis testing problem:

$\mathtt{H}_0$: $\widetilde{X}'$ was generated by Alice (Watermarked) ;
$\mathtt{H}_1$: $\widetilde{X}'$ was NOT generated by Alice (Unwatermarked) .

To address this problem, Alice will build a detector given

by

$$g(X;\mathcal{V}_\theta)=\begin{cases}1(\text{Watermarked}) & \text{if } \mathcal{V}_\theta(X)\geq\tau,\\ 0(\text{Unwatermarked}) & \text{if } \mathcal{V}_\theta(X)<\tau,\end{cases} \quad (1)$$

where $\tau\in\mathbb{R}$ is a threshold value and $\mathcal{V}_\theta$ (to be defined later) is a scoring function which takes the image input and output a value in $[0,1]$ to indicate its chance of being a sample generated by Alice.

Alice's Goals. Alice's objective is to design watermarking algorithms that fulfill the following objectives: **(1) Quality:** the quality of watermarked images should closely match that of the original, unwatermarked images; **(2) Identifiability:** both watermarked and unwatermarked content should be accurately distinguishable; **(3) Robustness:** the watermark should be resilient against various image manipulations.

Cathy's (Adversary) Goals. Cathy aims to design attack algorithms to meet the following objectives: **(1) Watermark Removal:** the watermarks embedded by Alice can be effectively eliminated after the attacks; **(2) Quality:** the attacks cannot significantly alter the images.

3. RAW

In this section, we formally introduce our RAW framework. At a high level, the RAW framework comprises two consecutive stages: a training stage and an inference stage. In the following subsection, we first provide an in-depth description of the training stage.

3.1. Training stage

Suppose Alice obtains a batch of diffusion model-generated images. The unwatermarked data are denoted as $\mathcal{D}^{\text{uwm}}\triangleq\{X_i\}_{i=1}^n$ for $i=1,\ldots,n$. Alice will need to embed watermarks into these images to protect intellectual property.

Definition 3.1 (Watermarking Module)**.** A watermarking module $\mathcal{E}_{\boldsymbol{w}}(\cdot)$ maps $\mathcal{X}$ to itself, with parameters $\boldsymbol{w}\in\mathbb{R}^{d_1}$.

The watermarking module can take the form of an encoder with an attention mechanism, as seen in the RivaGan (Zhang et al., 2019), or it can involve Fast Fourier Transformation (FFT) followed by frequency adjustments and an inverse FFT, as employed in DwtDct.

In our RAW framework, we propose to add two distinct watermarks into *both* frequency and spatial domains:

$$\mathcal{E}_{\boldsymbol{w}}(X)=\mathcal{F}^{-1}(\mathcal{F}(X)+c_1\times v)+c_2\times u, \quad (2)$$

where $v,u\in\mathcal{X}$ are two watermarks, $c_1,c_2>0$ determine the visibility of these watermarks, and $\mathcal{F}(\mathcal{F}^{-1})$ represents the Fast Fourier Transformation (FFT) (inverse FFT), respectively. For simplicity of notation, in the rest of this paper, we will denote $\boldsymbol{w}\triangleq[u,v]$.

The rationale for adopting the above approach is to simultaneously enjoy the distinct advantages offered by watermarks in both domains. In particular, the incorporation of watermark patterns in the frequency domain has been widely recognized for its effectiveness against certain image manipulations such as translations and resizing. Moreover, our empirical validation corroborates the improved robustness of spatial domain watermarking in the presence of noise perturbations.

We denote the watermarked dataset $\mathcal{E}_{\boldsymbol{w}}$ to be $\mathcal{D}^{\text{wm}}\triangleq\{\mathcal{E}_{\boldsymbol{w}}(X_i)\}_{i=1}^n$ for $i=1,\ldots,n$. Alice now wishes to distinguish the combined dataset $\mathcal{D}\triangleq\mathcal{D}^{\text{uwm}}\bigcup\mathcal{D}^{\text{wm}}$ with a verification module, which is a binary classifier.

Definition 3.2 (Verification Module)**.** A verification module is a mapping $\mathcal{V}_\theta(\cdot):\mathcal{X}\mapsto[0,1]$ parameterized by $\theta\in\mathbb{R}^{d_2}$.

The score generated by the verification module for an input image can be understood as the chance of this image being generated by Alice.

To fulfill Alice's first two goals, Alice will consider *jointly* training the watermarking and verification modules parameterized by $\boldsymbol{w}$ and θ, with the following loss function:

$$\text{BCE}(\mathcal{D})\triangleq\sum_{X\in\mathcal{D}}Y\log(\mathcal{V}_\theta(X))+(1-Y)\log(1-\mathcal{V}_\theta(X)), \quad (3)$$

where X is the training image and $Y\in\{0,1\}$ is the label indicting X is watermarked or not.

Recall that Alice also aims to enhance the robustness of the watermark algorithms. As a result, we consider transforming the combined datasets with different data augmentations $\mathcal{M}_1,\ldots,\mathcal{M}_m$ to obtain $\mathcal{D}^1\triangleq\mathcal{M}_1(\mathcal{D}),\ldots,\mathcal{D}^m\triangleq\mathcal{M}_m(\mathcal{D})$, respectively. Here, the data augmentations $\mathcal{M}_1,\ldots,\mathcal{M}_m$ are defined as follow.

Definition 3.3 (Modification Module)**.** An image modification module is a map $\mathcal{M}:\mathcal{X}\mapsto\mathcal{X}$.

To sum up, the overall loss function for our RAW framework is specified as:

$$\mathcal{L}_{\text{raw}}\triangleq\text{BCE}(\mathcal{D})+\sum_{k=1}^m\text{BCE}(\mathcal{D}^k), \quad (4)$$

where $\text{BCE}(\cdot)$ denotes the binary cross entropy loss as specified in Equation (3). The loss function above is composed of two terms: $\text{BCE}(\mathcal{D})$, which corresponds to the cross-entropy on the original combined datasets $\mathcal{D}$, and $\sum_{k=1}^m\text{BCE}(\mathcal{D}^k)$, signifying the cross-entropy on the augmented datasets $\mathcal{D}^1,\ldots,\mathcal{D}^m$. In our experiments, inspired by contrastive learning such as those presented in (Chen et al., 2020; Khosla et al., 2020), we adopt a two-view data augmentation approach by setting $m=2$.

3.1.1. Overall Training Algorithm

We describe the overall training algorithm below. For completeness, we provide the full pseudo-code as summarized in Algorithm 3 in the appendix. We consider conducting the following two steps alternatively.

- Optimize the verification module $\mathcal{V}_\theta$ based on the overall loss $\mathcal{L}_{\text{raw}}$ by stochastic gradient descent (SGD):

$$\theta_{t+1} \leftarrow \theta_t - \mu_t \nabla_\theta \mathcal{L}_{\text{raw}}(\theta_t, \boldsymbol{w}),$$

  where $\mu_t > 0$ is the step size at each step t.

- Optimize the watermark $\boldsymbol{w}$ based on $\mathcal{L}_0$ with sign-based stochastic gradient descent (SignSGD):

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \nu_t \operatorname{sign}\left(\nabla_{\boldsymbol{w}} \mathcal{L}_{\text{raw}}(\theta, \boldsymbol{w}_t)\right), \quad (5)$$

  where $\operatorname{sign}(\cdot)$ outputs the sign of each component, and $\nu_t > 0$ is the step size.

In the watermark update, Eq. (5), we opt for signSGD over vanilla SGD. This choice is motivated by several existing empirical observations that (sign-based) first-order methods could yield improved training and test performance in the context of data-level optimization problems in deep learning (Madry et al., 2017; Liu et al., 2021). Consequently, we utilize SignSGD for watermark optimization.

3.2. Inference Stage

We present a generic approach for Alice to obtain provable guarantees on the FPRs when using the previously trained $\mathcal{V}_\theta$ on test images, even amidst adversarial perturbations.

To begin with, we first examine a scenario where the future test data $X_{\text{test}} \in \mathcal{X}$ adheres to an IID pattern with the watermarked data $\mathcal{D}^{\text{wm}}$ generated by Alice, without undergoing any image modifications. In this case, Alice can employ conformal prediction to establish provable guarantees on the FPRs. The main idea is that, by utilizing the trained $\mathcal{V}_\theta$ as a scoring mechanism, the empirical quantile of the watermarked data's distribution will converge to the population counterpart. This convergence is guaranteed by the uniform convergence of cumulative distribution functions (CDFs). To be more specific, by setting the threshold τ, defined in Equation (1), to be the α-quantile (with finite-sample corrections) of predicted scores for watermarked data $\mathcal{V}_\theta(\mathcal{D}^{\text{wm}}) \triangleq \{\mathcal{V}_\theta(\mathcal{E}_{\boldsymbol{w}}(X_i))\}_{i=1}^n$, it can be shown (Lei & Wasserman, 2014) that the probability of the resulting detector g misclassifying a watermarked image X_{test} is upper bounded by α, with high probability, under the condition that X_{test} is IID with $\mathcal{D}^{\text{wm}}$. For completeness, a rigorous statement and its proof are provided in Appendix A.2.

The above argument assumes that the future test image X_{test} follows an IID pattern with the original watermarked data $\mathcal{D}^{\text{wm}}$. However, if the test image X_{test} undergoes manipulation or attack, denoted by $\mathcal{A}(X_{\text{test}})$, with $\mathcal{A} : \mathcal{X} \mapsto \mathcal{X}$ being an adversarial image manipulation, then it can deviate from the distribution of $\mathcal{D}^{\text{wm}}$. This deviation from IID will render the previous argument invalid. Moreover, in practice, Alice is unaware of the adversarial transformation $\mathcal{A}$ employed by the attacker, thus making it even more challenging to control the FPRs.

To address this problem, we propose to consider a robust version of the originally trained $\mathcal{V}_\theta$, denoted as $\mathcal{V}_{\tilde{\theta}}$, such that X and $\mathcal{A}(X)$ stay close under $\mathcal{V}_{\tilde{\theta}}$, namely

$$|\mathcal{V}_{\tilde{\theta}}(X) - \mathcal{V}_{\tilde{\theta}}(\mathcal{A}(X))| \leq \eta, \quad (6)$$

for all X and a small $\eta > 0$. The reason for finding such $\mathcal{V}_{\tilde{\theta}}$ is because we can relate $\mathcal{V}_{\tilde{\theta}}(\mathcal{A}(X_{\text{test}}))$ back to $\mathcal{V}_{\tilde{\theta}}(X_{\text{test}})$ which is IID with $\mathcal{V}_{\tilde{\theta}}(\mathcal{D}^{\text{wm}})$ (accessible to Alice) to establish the FPRs with previous arguments.

To develop the robust version from the base $\mathcal{V}_\theta$, we will build upon the following result from the randomized smoothing literature. Denote $\mathcal{N}(\mu, \Sigma)$ to be the normal distribution with mean μ and covariance Σ respectively, and $\Phi^{-1}(\cdot)$ to be the inverse of the cumulative distribution function of a standard normal distribution.

Lemma 3.4 ((Salman et al., 2019)). *Let $h : \mathbb{R} \to [0, 1]$ be a continuous function. Let $\sigma > 0$, and $H(X) = \mathop{\mathbb{E}}_{Z \sim \mathcal{N}(0, \sigma^2 I)}[h(X + Z)]$. Then the function $\Phi^{-1}(H(X))$ is σ^{-1}-Lipschitz with respect to ℓ_2 norm.*

The above result suggests that for *any* (continuous) base verification module (classifier) $\mathcal{V}_\theta$, we can obtain a smoothed version with

$$\mathcal{V}_{\tilde{\theta}}(X) = \Phi^{-1}\left(\mathop{\mathbb{E}}_{Z \sim \mathcal{N}(0, \sigma^2 I)}[\mathcal{V}_\theta(X + Z)]\right), \quad (7)$$

and it is guaranteed that $|\mathcal{V}_{\tilde{\theta}}(X) - \mathcal{V}_{\tilde{\theta}}(Y)| \leq \sigma^{-1}\|X - Y\|$, for any $X, Y \in \mathcal{X}$. Suppose the attacker employs an adversarial attack $\mathcal{A}$ such that $\|X - \mathcal{A}(X)\| \leq \gamma$. We have

$$|\mathcal{V}_{\tilde{\theta}}(X) - \mathcal{V}_{\tilde{\theta}}(\mathcal{A}(X))| \leq \frac{\gamma}{\sigma}. \quad (8)$$

Remark 3.5 ($\mathcal{A}$ can not be excessively adversarial). We emphasize that the transformation $\mathcal{A}$ should not be excessively adversarial. In other words, the parameter γ should be a very low value for both theoretical and practical reasons. From a theoretical perspective, an overly adversarial transformation $\mathcal{A}$ can result in trivial TPRs/FPRs. For instance, if watermarked images are transformed into a completely uniform all-white or all-black state, it becomes impossible to detect the watermark. From a practical standpoint, an excessively adversarial transformation $\mathcal{A}$ tends to overwrite the original content within the images. This directly contradicts the intentions of attackers and may not achieve the desired stealthy modifications.

3.2.1. Overall Inference Algorithm

Given a pair of $(\mathcal{E}_{\boldsymbol{w}}, \mathcal{V}_\theta)$, a desired robust range $\gamma > 0$, and a smoothing parameter $\sigma > 0$, Alice now will set the thresholding value τ, as introduced in Equation (1), to satisfy:

$$\hat{F}\big(\tau + \frac{\gamma}{\sigma}\big) = \alpha - \sqrt{(\log(2/\delta)/(2n))}, \tag{9}$$

where $\delta \in (0, 1)$ is a violation rate describing the probability that the FPRs exceeds α, and $\hat{F}$ is the empirical cumulative distribution function of $\{\mathcal{V}_{\tilde{\theta}}(\mathcal{E}_{\boldsymbol{w}}(X_i))\}_{i=1}^n$, where

$$\mathcal{V}_{\tilde{\theta}}(\mathcal{E}_{\boldsymbol{w}}(X_i)) \triangleq \Phi^{-1}\bigg(\mathop{\mathbb{E}}_{Z\sim\mathcal{N}(0,\sigma^2 I)}[\mathcal{V}_\theta(\mathcal{E}_{\boldsymbol{w}}(X_i) + Z)]\bigg).$$

The next result shows that if a future test input comes from the same distribution as the watermarked data $\mathcal{D}^{\text{wm}}$, the above procedure can be configured to achieve any prespecified false positive rate α with high probability.

Theorem 3.6 (Certified FPRs of g based on threshold in Equation (9))**.** *Given any watermarked dataset $\mathcal{D}^{wm}$ and its associated verification module $\mathcal{V}_\theta$, suppose that the test data (X_{test}, Y_{test}) are IID drawn from the distribution of $\mathcal{D}^{wm}$. Given any $\delta \in (0, 1)$ and $\gamma > 0$, for any (adversarial) image transformations $\mathcal{A}$ such that $\|\mathcal{A}(X) - X\| \leq \gamma$ for all $X \in \mathcal{X}$, the detector $g(\cdot)$ introduced in Equation (1), with the threshold τ as specified in Equation (9) satisfies*

$$\mathbb{P}\bigg(g(\mathcal{A}(X_{test})) = 0\,(\textit{Unwatermarked}) \mid \mathcal{D}^{wm}\bigg) \leq \alpha,$$

with probability at lease $1 - \delta$ for any $\alpha \in (0, 1)$ such that $\alpha > \sqrt{(\log(2/\delta)/(2n))}$.

Due to space constraints, the proof is provided in Appendix A.1. The above result shows that by using the decision rule as specified in Equation (9), Alice can obtain a provable guarantee on the Type I error rate in terms of detecting future test input X_{test} even X_{test} is adversarially perturbed within γ-range (as measured by ℓ_2-norm), under the condition that the future test input X_{test} is independently and identically distributed as the $\mathcal{D}^{\text{wm}}$, namely watermarked samples generated by the artist.

4. Experiments

In this section, we conduct a comprehensive evaluation of our proposed RAW, including (1) detection performance, (2) robustness, (3) watermarking speed, (4) the quality of watermarked images, and (5) the provable FPRs guarantees under adversarial attacks. Our findings reveal significantly enhanced robustness in RAW while preserving the quality of generated images. Furthermore, a substantial reduction in watermark injection time, indicates the suitability of RAW for on-the-fly deployment. All the experiments were conducted on machines equipped with Nvidia Tesla A100s.

4.1. Experimental setups

Datasets (1) In line with the previous work (Wen et al., 2023), we employ Stable Diffusion-v2-1 (Rombach et al., 2022), an open-source text-to-image diffusion model, with DDIM sampler, to generate images. All the prompts used for image generation are sourced from the MS-COCO dataset (Lin et al., 2014). **(2)** We further evaluate our RAW utilizing DBdiffusion (Wang et al., 2022b), a dataset consisting of 14 million images generated by Stable Diffusion. This dataset encompasses a wide array of images produced under various prompts, samplers, and user-defined hyperparameters. Ablation studies examining various generative models, such as SDXL-1.0, are detailed in Appendix D.

Verification Modules/Classifiers In terms of verification modules, for all the results reported in the main text, we utilize ResNet 18 (He et al., 2016). For training the verification modules, 500 images are randomly selected for training for each dataset. Subsequently, we evaluate the trained watermarks and associated models on 1000 new, unwatermarked images and their watermarked versions. Ablation studies on using different models, such as VGG (Simonyan & Zisserman, 2014) as verification modules, and different number of training data are provided in Section 4.4 and Appendix D.

Watermark Parameters In Equation (2) of RAW, two parameters c_1 and c_2 control the invisibility of the watermark, thereby influencing the quality of watermarked images. In the main text results, we set $c_1 = c_2 = 0.05$ to align with the image quality of watermarked images produced by other state-of-the-art methods (refer to the 'FID' and 'CLIP' columns in Table 2). We conduct an ablation study exploring different values of c_1 and c_2 in Section 4.4 and Appendix D. For all other watermarking techniques, we implement them using the open-source package employed by the Official Stable Diffusion Model.

Evaluation Metrics (1) To assess the detection performance, we adhere to the convention of reporting the area under the curve of the receiver operating characteristic (AUROC) (Wen et al., 2023; Fernandez et al., 2022; 2023). **(2)** For evaluating the quality of the watermarked images, we adopt both the Frechet Inception Distance (FID) (Heusel et al., 2017) and the CLIP score (Radford et al., 2021), following the methodology outlined in (Wen et al., 2023). All metrics are averaged across 5 independent runs.

4.2. Main Results

Detection performance and image generation quality To ensure a fair comparison, we primarily evaluate our proposed RAW against other model-agnostic approaches, presenting the summarized results in Table 2, along with visual examples illustrated in Figure 2. Additionally, for completeness, we compare our proposed RAW against model-specific

Table 2: Summary of main results. 'AUROC (Nor)' denotes the AUROC performance without image manipulations or adversarial attacks. 'AUROC (Adv)' represents the average performance across nine distinct image manipulations and attacks. The 'Encoding Speed' column denotes the efficiency of injecting watermarks into images post-training, measured in seconds (CPU time) per image.

Dataset	Method	Model Agnostic	AUROC (Nor) ↑	AUROC (Adv) ↑	FID ↓	CLIP ↑	Encoding Speed ↓
MS-Coco FID: 24.12 CLIP: 0.382	DwTDcT	✓	0.83	0.54	25.10	0.359	0.048
	DwTDcTSvd	✓	0.98	0.75	25.21	0.361	0.122
	RivaGan	✓	0.99	0.81	24.87	0.359	1.16
	Stegastamp	✓	0.99	0.93	42.31	0.291	1.45
	RAW (Ours)	✓	0.98	**0.92**	24.75	0.360	**0.0051**
DBdiffusion FID: 4.56 CLIP: 0.412	DwTDcT	✓	0.81	0.55	4.63	0.427	0.048
	DwTDcTSvd	✓	0.99	0.78	4.61	0.421	0.110
	RivaGan	✓	0.99	0.82	4.82	0.424	1.87
	Stegastamp	✓	0.99	0.92	7.34	0.386	1.90
	RAW (Ours)	✓	0.98	**0.90**	5.17	0.425	**0.0078**

Table 3: AUROC performance of state-of-the-art methods under 9 (adversarial) image manipulations: Rotation 90°, Cropping and resizing 70%, Gaussian Blur with a kernel size of $(7, 9)$ and bandwidth of 4, Noise with IID mean Gaussian $\sigma = 0.05$, Jitter with brightness factor 0.6, JPEG compression with quality 50, and 3 attacks for removing watermarks.

Datasets	MS-COCO				DBDiffusion			
	DwtDct	DwtDctSvd	RivaGan	RAW (Ours)	DwtDct	DwtDctSvd	RivaGan	RAW (Ours)
JPEG 50	0.612	0.995	0.996	0.914	0.503	0.954	0.997	0.999
Rotation 90°	0.508	0.547	0.391	0.956	0.471	0.541	0.381	0.824
Cropping 70%	0.640	0.521	0.990	0.957	0.651	0.613	0.991	0.843
Gaussian Blur	0.524	0.916	0.999	0.936	0.533	0.994	0.999	0.999
Gaussian Noise	0.475	0.763	0.999	0.902	0.844	0.988	0.999	0.999
Jittering	0.651	0.782	0.987	0.956	0.467	0.688	0.987	0.999
VAE Att1	0.502	0.728	0.628	0.895	0.488	0.751	0.673	0.801
VAE Att2	0.483	0.775	0.671	0.912	0.498	0.725	0.630	0.810
Diff Att	0.498	0.713	0.698	0.828	0.507	0.733	0.703	0.824
Average	0.543	0.748	0.817	**0.918**	0.551	0.776	0.818	**0.901**

methods and provide the results in the appendix. Our RAW exhibits comparable performance to encoder-decoder-based approaches, e.g., RivaGAN, while concurrently achieving similar FID and CLIP scores, which underscores superior image quality compared to alternatives.

Robust detection performance We assess the robustness of our proposed RAW against six common data augmentations and three adversarial attacks in this subsection. The data augmentation set comprises: color jitter with a brightness factor of 0.5, JPEG compression with quality 50, rotation by 90°, addition of Gaussian noise with 0 mean and standard deviation 0.05, Gaussian blur with a kernel size of $(7, 9)$ and bandwidth 4, and 70% random cropping and resizing. For adversarial attacks, we select three state-of-the-art methods for removing watermarks, with two VAE-based attacks Bmshj2018 (Ballé et al., 2018) (VAE Att1) and Cheng2020 (Cheng et al., 2020) (VAE Att2) from CompressAI (Bégaint et al., 2020), and one diffusion-model attacks.

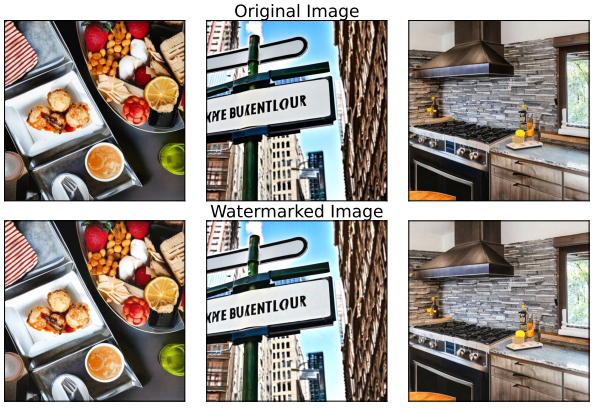

Figure 2: Examples of RAW-watermarked images (bottom row). More visual examples are given in Appendix D.5.

The averaged results are in the 'AUROC (Adv)' column of Table 2 and the detailed results are summarized in Table 3. Our approach demonstrates superior performance compared with alternative methods. Specifically, across both datasets, the average AUROC for our RAW increased by 70% and 13% for nine image manipulations/attacks, surpassing frequency- and encoder-decoder-based methods. We note that Stegastamp demonstrates similar averaged robust detection performance compared to ours. However, this comes at the expense of *significantly reduced image quality*, as evidenced by markedly increased FID scores.

Watermark injection speed We investigate the time costs needed to embed watermarks into images. We note that the watermark injection process occurs post-training. Therefore, our watermark injections only necessitate one FFT, two additions, and another inverse FFT. In Table 4, we present CPU time (in seconds) elapsed for injecting watermarks into different image quantities. Notably, our method achieves substantial time efficiency improvements, approximately $\mathbf{30}\times$ ($\mathbf{200}\times$) faster than the frequency-based (encoder-decoder based) method, respectively. This is attributed to streamlined batch operations in our RAW. This highlights the suitability of our approach for on-the-fly deployment.

Table 4: CPU time (seconds) elapsed for injecting watermarks into images. **Lower** values are preferred.

Batch Size $\rightarrow$	5 images	100 images	500 images
DwtDct	0.27	4.8	24.5
DwtDctSvd	0.64	12.2	60.1
RivaGAN	5.52	116	> 500
Stegastamp	7.34	134	> 500
RAW (Ours)	0.35	0.51	0.76

4.3. Certified FPRs

We assess the certified FPRs performance of our proposed RAW by varying the FPRs rate α pre-specified by Alice. We set the adversaril radius $\gamma = 0.001$ and the smoothing parameter $\sigma = 0.05$. We summarize the results of five independent runs in Figure 4(a) and report the mean (with standard error < 0.002). The results show that the FPRs of RAW consistently matches the theoretical upper bounds (i.e., α), supporting the result presented in Theorem 3.6.

4.4. Ablation Studies

Trade-off between robustness and image quality We explore the trade-off between robustness and image quality by adjusting the watermark strength parameters c_1 ($= c_2$), as illustrated in Figure 3 below. We note that with increasing values of c_1 and c_2, the average AUROC, under 9 (adversarial) images manipulations, also increases, while the image quality only exhibits a slight degradation, as indicated by the slightly increased FID value. These findings also highlight the stability of the watermark hyperparameters c_1 and c_2 in our proposed RAW, a desirable characteristic for real-world deployments, as preferred by practitioners.

Effect of training sample size on prediction performance We manipulate the sample size of the watermarked training dataset $\mathcal{D}^{\text{wm}}$ to assess its impact on detection performance. These findings are illustrated in Figure 4(b), where we note that satisfactory detection performance can be achieved with a reasonably small training dataset, e.g., 100 images.

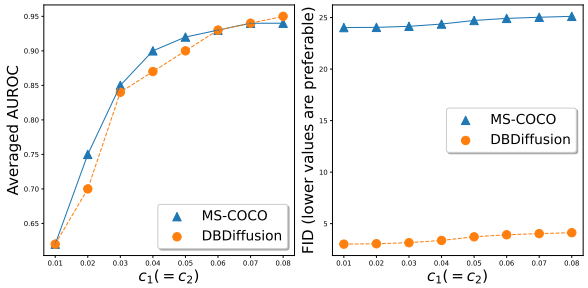

Figure 3: The tradeoff between the quality of watermarked images, assessed using FID (lower values are preferable), and the detection robustness.

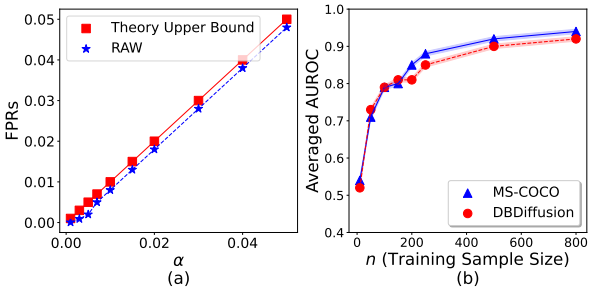

Figure 4: (a) FPRs of our proposed RAW (b) Impact of training sample size on detection performance.

5. Conclusion

In this study, we introduce the RAW framework as a versatile watermarking approach essential for protecting intellectual property and mitigating potential misuse of AI-generated images. The proposed RAW framework offers several notable features, including significantly enhanced watermark encoding speed and/or detection performance, along with the assurance of provable guarantees on false positive rates even under adversarial perturbations in test images. Experimental findings across various datasets validate its advantages.

Several promising research directions include investigating the maximum number of concurrent watermarks learnable in a single training session and optimal smoothing strategies for wider certified radii. The Appendix contains proofs, experimental details, and ablation studies.

Broader Impact

This paper aims to contribute to the advancement of trustworthy machine learning, particularly in ensuring the safe and legitimate use of contemporary generative artificial intelligence. Our efforts could have several positive societal implications, such as protecting intellectual property and preventing potential misuse of AI-generated images.

References

Adi, Y., Baum, C., Cisse, M., Pinkas, B., and Keshet, J. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX Security Symposium (USENIX Security 18)*, pp. 1615–1631, 2018.

Altun, H. O., Orsdemir, A., Sharma, G., and Bocko, M. F. Optimal spread spectrum watermark embedding via a multistep feasibility formulation. *IEEE transactions on image processing*, 18(2):371–387, 2009.

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

Bégaint, J., Racapé, F., Feltman, S., and Pushparaja, A. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020.

Chen, P.-Y. Model reprogramming: Resource-efficient cross-domain machine learning. *arXiv preprint arXiv:2202.10629*, 2022.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Cheng, Z., Sun, H., Takeuchi, M., and Katto, J. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7939–7948, 2020.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pp. 1310–1320. PMLR, 2019.

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

Cox, I. J., Kilian, J., Leighton, T., and Shamoon, T. Secure spread spectrum watermarking for images, audio and video. In *Proceedings of 3rd IEEE international conference on image processing*, volume 3, pp. 243–246. IEEE, 1996.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Duchi, J. C., Bartlett, P. L., and Wainwright, M. J. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

Fernandez, P., Sablayrolles, A., Furon, T., Jégou, H., and Douze, M. Watermarking images in self-supervised latent spaces. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3054–3058. IEEE, 2022.

Fernandez, P., Couairon, G., Jégou, H., Douze, M., and Furon, T. The stable signature: Rooting watermarks in latent diffusion models. *arXiv preprint arXiv:2303.15435*, 2023.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gu, T., Dolan-Gavitt, B., and Garg, S. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Gupta, G., Gupta, V., and Chandra, M. Review on video watermarking techniques in spatial and transform domain. In *Information Systems Design and Intelligent Applications: Proceedings of Third International Conference INDIA 2016, Volume 2*, pp. 683–691. Springer, 2016.

Hartung, F. and Girod, B. Watermarking of uncompressed and compressed video. *Signal processing*, 66(3):283–301, 1998.

Hayes, J. and Danezis, G. Generating steganographic images via adversarial training. *Advances in neural information processing systems*, 30, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hernandez, J. R., Amado, M., and Perez-Gonzalez, F. Dct-domain watermarking techniques for still images: Detector performance analysis and a new structure. *IEEE transactions on image processing*, 9(1):55–68, 2000.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and 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.

Kandi, H., Mishra, D., and Gorthi, S. R. S. Exploring the learning capabilities of convolutional neural networks for robust image watermarking. *Computers & Security*, 65: 247–268, 2017.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Kim, C., Min, K., Patel, M., Cheng, S., and Yang, Y. Wouaf: Weight modulation for user attribution and fingerprinting in text-to-image diffusion models. *arXiv preprint arXiv:2306.04744*, 2023.

Lei, J. and Wasserman, L. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(1): 71–96, 2014.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and 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.

Liu, F., Han, B., Liu, T., Gong, C., Niu, G., Zhou, M., Sugiyama, M., et al. Probabilistic margins for instance reweighting in adversarial training. *Advances in Neural Information Processing Systems*, 34:23258–23269, 2021.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Müller, R., Kornblith, S., and Hinton, G. E. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

Nie, G., Kim, C., Yang, Y., and Ren, Y. Attributing image generative models using latent fingerprints. *arXiv preprint arXiv:2304.09752*, 2023.

O'Ruanaidh, J. J. and Pun, T. Rotation, scale and translation invariant digital image watermarking. In *Proceedings of International Conference on Image Processing*, volume 1, pp. 536–539. IEEE, 1997.

Pereira, S., Ruanaidh, J. J. O., Deguillaume, F., Csurka, G., and Pun, T. Template based recovery of fourier-based watermarks using log-polar and log-log maps. In *Proceedings IEEE international conference on multimedia computing and systems*, volume 1, pp. 870–874. IEEE, 1999.

Pitas, I. A method for watermark casting on digital image. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(6):775–780, 1998.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents, 2022. *URL https://arxiv. org/abs/2204.06125*, 7, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and 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.

Sag, M. Copyright safety for generative ai. *Forthcoming in the Houston Law Review*, 2023.

Salman, H., Li, J., Razenshteyn, I., Zhang, P., Zhang, H., Bubeck, S., and Yang, G. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Tancik, M., Mildenhall, B., and 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.

Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pp. 601–618, 2016.

Uchida, Y., Nagai, Y., Sakazawa, S., and Satoh, S. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on international conference on multimedia retrieval*, pp. 269–277, 2017.

Vapnik, V. N. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

Verdoliva, L. Media forensics and deepfakes: an overview. *IEEE Journal of Selected Topics in Signal Processing*, 14 (5):910–932, 2020.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

Wang, G., Xian, X., Srinivasa, J., Kundu, A., Bi, X., Hong, M., and Ding, J. Demystifying poisoning backdoor attacks from a statistical perspective. In *International Conference on Learning Representations (ICLR)*, 2024.

Wang, Q. Invisible watermark. `https://github.com/ShieldMnt/invisible-watermark`, 2022.

Wang, Q., Liu, F., Zhang, Y., Zhang, J., Gong, C., Liu, T., and Han, B. Watermarking for out-of-distribution detection. *Advances in Neural Information Processing Systems*, 35:15545–15557, 2022a.

Wang, Z. J., Montoya, E., Munechika, D., Yang, H., Hoover, B., and Chau, D. H. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv preprint arXiv:2210.14896*, 2022b.

Wen, Y., Kirchenbauer, J., Geiping, J., and Goldstein, T. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *arXiv preprint arXiv:2305.20030*, 2023.

Wikipedia contributors. Stable diffusion, 2023. URL `https://en.wikipedia.org/wiki/Stable_Diffusion`.

Xian, X., Hong, M., and Ding, J. Understanding model extraction games. In *2022 IEEE 4th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*, pp. 285–294. IEEE, 2022.

Xian, X., Wang, G., Srinivasa, J., Kundu, A., Bi, X., Hong, M., and Ding, J. A unified detection framework for inference-stage backdoor defenses. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023a.

Xian, X., Wang, G., Srinivasa, J., Kundu, A., Bi, X., Hong, M., and Ding, J. Understanding backdoor attacks through the adaptability hypothesis. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023b.

Zhang, K. A., Xu, L., Cuesta-Infante, A., and Veeramachaneni, K. Robust invisible video watermarking with attention. *arXiv preprint arXiv:1909.01285*, 2019.

Zhao, Y., Pang, T., Du, C., Yang, X., Cheung, N.-M., and Lin, M. A recipe for watermarking diffusion models. *arXiv preprint arXiv:2303.10137*, 2023.

Zhu, J., Kaplan, R., Johnson, J., and Fei-Fei, L. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 657–672, 2018.

Appendix for RAW: A Robust and Agile Plug-and-Play Watermark Framework for AI-Generated Images with Provable Guarantees

In Section A, we outline the formal proofs supporting the theoretical findings introduced in the main text. In Section B, we present omitted details, encompassing pseudocode and additional deliberations regarding the construction of RAW. In Section C, we offer implementation details of our experiments. In Section D, we present additional ablation studies on different hyperparameters.

A. Proof of theoretical results

A.1. Proof of Theorem 3.6

In this section, we provide the proof for Theorem 3.6. To begin with, we first revisit the overall inference procedure employed by our RAW and provide several detailed discussions. For the reader's convenience, we include the pseudo-code outlining the inference stage protocol of RAW in Algorithm 1 below.

Algorithm 1 Conformal (Inference) Watermark Detection

Input: querying input X_{test}, watermarked dataset $D^{\text{wm}} = \{\mathcal{E}_{\boldsymbol{w}}(X_i)\}_{i=1}^n$, verification module $\mathcal{V}_\theta$, desired false positive rate $\alpha \in (0,1)$, violation rate $\delta \in (0,1)$, desired adversarial robust range $\gamma > 0$, smoothing parameter $\sigma > 0$, and the smoothed/robust verification module $\mathcal{V}_{\tilde{\theta}}(\cdot) \triangleq \Phi^{-1}(\mathbb{E}_{Z\sim\mathcal{N}(0,\sigma^2 I)}[\mathcal{V}_\theta(\mathcal{E}_{\boldsymbol{w}}(\cdot)+Z)])$

1: Receiving a future query sample X_{test}
2: **for** $i = 1$ to n **do**
3: Calculate $s_i \triangleq \mathcal{V}_{\tilde{\theta}}(\mathcal{E}_{\boldsymbol{w}}(X_i))$ // $X_i \in D^{\text{wm}}$
4: **end for**
5: Select the decision threshold τ according to Equation (10).
6: Determine if X_{test} is watermarked if $\mathcal{V}_{\tilde{\theta}}(\mathcal{E}_{\boldsymbol{w}}(X_{\text{test}})) \geq \tau$

Output: The decision if the sample X_{test} is a watermarked sample or not

The overall inference process for the proposed RAW mainly involves determining a decision threshold value τ. This threshold is calculated based on the provided watermarked dataset D^{wm} and the corresponding (smoothed) verification module $\mathcal{V}_{\tilde{\theta}}$, and it should satisfy the following condition:

$$\hat{F}(\tau + \frac{\gamma}{\sigma}) = \alpha - \sqrt{\frac{\ln\frac{2}{\delta}}{2n}}, \tag{10}$$

where $\delta \in (0,1)$ is the violation rate describing the probability that the (FPRs) exceeds α, and $\hat{F}$ is the empirical cumulative distribution function of the watermarked dataset under $\mathcal{V}_{\tilde{\theta}}$, i.e., $\{\mathcal{V}_{\tilde{\theta}}(\mathcal{E}_{\boldsymbol{w}}(X_i))\}_{i=1}^n$. In the case where $\sqrt{(\log(2/\delta)/(2n))} > \alpha$, we set the thresholding value τ to be the maximum of $\{\mathcal{V}_{\tilde{\theta}}(\mathcal{E}_{\boldsymbol{w}}(X_i))\}_{i=1}^n$. We note that under such cases there will be no theoretical guarantees in term of FPRs.

The proof of Theorem 3.6 relies on a crucial concept related to the convergence property of empirical cumulative distribution functions (ECDFs). Specifically, it asserts that by employing a suitable score function, e.g., the robust/smoothed verification module $\mathcal{V}_{\tilde{\theta}}$, the empirical rank or quantile of the distribution will eventually approach the population counterpart. This convergence is ensured by the uniform convergence of cumulative distribution functions (CDFs). Consequently, to establish the proof for Theorem 1, we will first present the subsequent outcome, which accurately measures the uniform convergence of CDFs.

Lemma A.1 (Dvoretzky–Kieffer–Wolfowitz inequality)**.** *Given a natural number n, let $X_1, X_2, \ldots, X_n$ be real-valued independent and identically distributed random variables with cumulative distribution function $F(\cdot)$. Let $\hat{F}(\cdot)$ denote the associated empirical distribution function.*

The interval that contains the true CDF, $F(x)$, with probability $1-\delta$ is specified as

$$\hat{F}(x) - \varepsilon \leq F(x) \leq \hat{F}(x) + \varepsilon \text{ where } \varepsilon = \sqrt{\frac{\ln\frac{2}{\delta}}{2n}}.$$

In the following, we present the formal proof of Theorem 3.6 in the main text.

Proof. Note that X_{test} is IID drawn from the watermarked data distribution and we have

$$\begin{aligned}
&\mathbb{P}(g(\mathcal{A}(X_{\text{test}})) = 0 \text{ (Unwatermarked)} \mid \mathcal{D}^{\text{wm}}) \\
&= \mathbb{P}(\mathcal{V}_{\tilde{\theta}}(\mathcal{A}(X_{\text{test}})) \leq \tau \mid \mathcal{D}^{\text{wm}}) && (11)\\
&\leq \mathbb{P}(\mathcal{V}_{\tilde{\theta}}(X_{\text{test}}) - \frac{\gamma}{\sigma} \leq \tau \mid \mathcal{D}^{\text{wm}}) && (12)\\
&= \mathbb{E}_{X_{\text{test}}} \mathbf{1}\{\mathcal{V}_{\tilde{\theta}}(X_{\text{test}}) \leq \tau + \frac{\gamma}{\sigma} \mid \mathcal{D}^{\text{wm}}\} \\
&= \mathbb{E}_{X_{\text{test}}} \mathbf{1}\{F(\mathcal{V}_{\tilde{\theta}}(X_{\text{test}})) \leq F(\tau + \frac{\gamma}{\sigma}) \mid \mathcal{D}^{\text{wm}}\} && (13)\\
&= \mathbb{P}(F(\mathcal{V}_{\tilde{\theta}}(X_{\text{test}})) \leq F(\tau + \frac{\gamma}{\sigma}) \mid \mathcal{D}^{\text{wm}}) \\
&\leq \mathbb{P}(F(\mathcal{V}_{\tilde{\theta}}(X_{\text{test}})) \leq \hat{F}(\tau + \frac{\gamma}{\sigma}) + \varepsilon \mid \mathcal{D}^{\text{wm}}) \quad (\varepsilon = \sqrt{\frac{\ln \frac{2}{\delta}}{2n}}) && (14)\\
&= \alpha - \varepsilon + \varepsilon && (15)\\
&= \alpha,
\end{aligned}$$

holds with probability at least $1 - \delta$. The equation (11) is because of the decision rule as specified in Algorithm 1, and the inequality (12) is due to the lipschitz condition of $\mathcal{V}_{\tilde{\theta}}$ with parameter σ^{-1}, namely

$$|\mathcal{V}_{\tilde{\theta}}(X) - \mathcal{V}_{\tilde{\theta}}(Y)| \leq \sigma^{-1}\|X - Y\|,$$

for any $X, Y \in \mathcal{X}$. Additionally, the F in equation (13) represents the CDF of the watermarked data under $\mathcal{V}_{\tilde{\theta}}(\cdot)$, i.e., $\mathcal{V}_{\tilde{\theta}}(\mathcal{E}_{\boldsymbol{w}}(X))$, while $\hat{F}$ in (14) denotes the empirical CDF obtained from $\mathcal{D}^{\text{wm}}$, i.e., $\{\mathcal{V}_{\tilde{\theta}}(\mathcal{E}_{\boldsymbol{w}}(X_i))\}_{i=1}^n$ under $\mathcal{V}_{\tilde{\theta}}(\cdot)$. The inequality in (14) arises from the DKW inequality as specified in Lemma A.1. Furthermore, the equation (15) is based on the fact that the CDF follows a uniform distribution (a result of the probability integral transformation) and the selection of the thresholding value specified in Equation (10). □

A.2. Provable FPRs without adversarial attacks

In this section, we present the omitted results of the provable FPRs concerning the watermark detection performance of our RAW in the absence of adversarial attacks, as elaborated in Line 225 of the main text.

As there are no anticipated adversarial attacks on test images, there is consequently no requirement to apply smoothing/robustification to the trained verification model $\mathcal{V}_\theta$. The corresponding pseudo-code is outlined in Algorithm 2 below. Similarly, the updated thresholding value τ is selected to satisfy the condition.

Algorithm 2 Conformal (Inference) Watermark Detection under no adversarial attacks

Input: querying input X_{test}, watermarked dataset $D^{\text{wm}} = \{(\mathcal{E}_{\boldsymbol{w}}(X_i))\}_{i=1}^n$, verification module $\mathcal{V}_\theta$, desired false positive rate $\alpha \in (0, 1)$, violation rate $\delta \in (0, 1)$

1: Receiving a future query sample X_{test}
2: **for** $i = 1$ to n **do**
3: Calculate $s_i \triangleq \mathcal{V}_\theta(\mathcal{E}_{\boldsymbol{w}}(X_i))$ // $X_i \in D^{\text{wm}}$
4: **end for**
5: Select the decision threshold τ according to Equation (10).
6: Determine if X_{test} is watermarked if $\mathcal{V}_\theta(\mathcal{E}_{\boldsymbol{w}}(X_{\text{test}})) \geq \tau$

Output: The decision if the sample X_{test} is a watermarked sample or not

$$\hat{F}(\tau) = \alpha - \sqrt{\frac{\ln \frac{2}{\delta}}{2n}}, \tag{16}$$

where $\delta \in (0,1)$ is the violation rate describing the probability that the (FPRs) exceeds α, and $\hat{F}$ is the empirical cumulative distribution function of the watermarked dataset under $\mathcal{V}_\theta$, i.e., $\{\mathcal{V}_\theta(\mathcal{E}_{\boldsymbol{w}}(X_i))\}_{i=1}^n$. Contrasting the selection of τ under adversarial attacks in Equation (10), we note that the term γ/σ is omitted because of the absence of adversarial attacks. In the case where $\sqrt{(\log(2/\delta)/(2n))} > \alpha$, we set the thresholding value τ to be the maximum of $\{\mathcal{V}_\theta(\mathcal{E}_{\boldsymbol{w}}(X_i))\}_{i=1}^n$.

Theorem A.2 (Certified FPRs under no adversarial attacks)**.** *For any watermarked dataset $\mathcal{D}^{wm}$ and its associated verification module $\mathcal{V}_\theta$, suppose that the test data (X_{test}, Y_{test}) are IID drawn from the distribution of $\mathcal{D}^{wm}$. Given $\delta \in (0,1)$, the detector$g(\cdot)$ (defined in Line 231 in the main text) with the threshold τ as specified in Equation (16) satisfies*

$$\mathbb{P}(g(X_{test})) = 0 \text{ (Unwatermarked)} \mid \mathcal{D}^{wm}) \leq \alpha$$

with probability at least $1-\delta$ for any $\alpha \in (0,1)$ such that $\alpha > \sqrt{(\log(2/\delta)/(2n))}$.

Proof. The proof follows the same approach as the proof for Theorem 1, with the exclusion of the γ/σ term. □

B. Omitted Details and Further Discussions

In this section, we initially present the omitted pseudo-code for the training algorithm, followed by further discussions regarding the design of RAW.

B.1. Pseudocode for training algorithms

The pseudocode for the overall training pipeline of RAW is outlined in Algorithm 3.

Algorithm 3 Training Algorithms for RAW

Input: (I) Image sets generated from a diffusion model $\{X_i\}_{i=1}^n$; (II) watermark visibility parameter c_1, c_2; (III) learning rates $\{\mu_t\}_{t=1}^T, \{\nu_t\}_{t=1}^T$.
Initialize: (1) a verification module $\mathcal{V}_\theta : \mathcal{X} \mapsto [0,1]$, (2) a watermarking module: $\mathcal{E}_{\boldsymbol{w}}(X) = \mathcal{F}^{-1}(\mathcal{F}(X) + c_1 \times v) + c_2 \times w$ with each entries in $u, v \in \mathcal{X}$ initialized as IID uniform random variables.

1: **for** $i = 1$ to T **do**
2: Clipping the watermarked data to be within the range $[0,1]$;
3: Given $\mathcal{V}_\theta$, optimizing $\boldsymbol{w}$ based on $\mathcal{L}_{\text{raw}}$ with SignSGD;
4: Given the watermark $\boldsymbol{w}$, updating θ based on $\mathcal{L}_{\text{raw}}$ with SGD;
5: **end for**

Output: (1) The verification module $\mathcal{V}_\theta$; (2) Watermarking method $\mathcal{E}_{\boldsymbol{w}}$

B.2. Further Discussions

We now elaborate on two pivotal aspects of our watermark designs and overarching training algorithms: **(I)** the joint training scheme for watermarking and verification modules, and **(II)** the integration of spatial-domain watermarks.

(I) The joint training scheme for watermarking and verification modules. Theoretically, using standard arguments from classical learning theory (Vapnik, 1999), it can be shown that training both the watermarking and the verification modules to distinguish between watermarked and unwatermarked data will not lead to a test accuracy worse than when the watermark is fixed, and only the model is trained. From a practical perspective, the initially randomly initialized watermarks may not align well with specific training data, emphasizing the need to optimize watermarks for distinct data scenarios. Our empirical observations support this notion, as evidenced in Figure 5 (left), where the joint training scheme leads to a significantly higher test accuracy and lower training loss compared with the scenario where the watermark is fixed.

(II) The inclusion of spatial domains. Classical methods for embedding watermarks primarily introduce them into the frequency domains of images (Cox et al., 2007). However, it has been empirically observed that such watermarks are susceptible to manipulations, such as Gaussian noise (Wen et al., 2023). To overcome this vulnerability, we draw inspiration from the model reprogramming literature (Chen, 2022), where watermarks are incorporated into the spatial domain to enhance accuracy in distinguishing in- and out-distribution data (Wang et al., 2022a). Consequently, we explore the integration of watermarks into the spatial domain (in addition to the frequency domain), as outlined in Equation (2). We empirically observed that including spatial watermarks could significantly boost the test accuracy of the trained verification module under Gaussian-noise manipulations on test data, as depicted in Figure 5 (right).

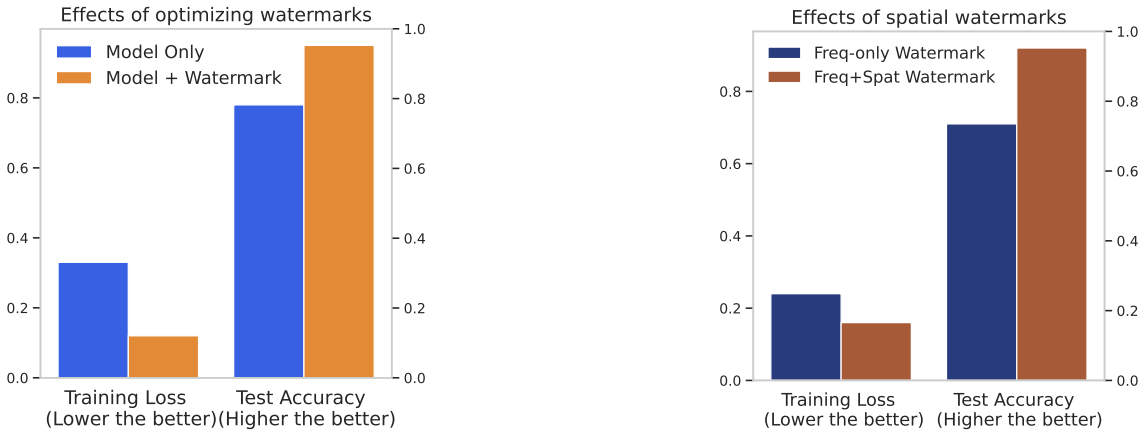

Figure 5: Effects of (1) joint training and (2) spatial watermarks.

C. Implementation details

In this section, we first list the implementation details for the results presented in the main text. Next, we present ablation studies on the performance of our RAW by using different hyper-parameters.

C.1. Implementation Details

Watermark setup Recall that the watermarking module in our RAW takes the form of

$$\mathcal{E}_{\boldsymbol{w}}(X) = \mathcal{F}^{-1}(\mathcal{F}(X) + c_1 \times u) + c_2 \times v,$$

where $u, v \in \mathcal{X}$ are two watermarks injected into spatial and frequency domains, respectively. For the results presented in the main text, we set the watermark strength parameters c_1 and c_2 to be 0.05 each. To enhance invisibility further, we implement a circular mask with a radius of 200 for the frequency domain watermarks, inspired by the procedure outlined in (Wen et al., 2023).

Verification module setup In terms of the verification module $\mathcal{V}_\theta$, throughout all the experiments in the main text, we adopt the pre-trained ResNet 18 (He et al., 2016) architecture.

Training data augmentation Two types of data augmentations are employed during the training phase. (1) To enhance the robustness of the trained verification model $\mathcal{V}_\theta$ against Gaussian noise, ensuring its sustained predictive effectiveness post-smoothing, we introduce Gaussian noise into the training data. This augmentation entails the addition of noise with a mean of zero and a standard deviation of 0.5 during the training process. (2) Additionally, we employ standard image augmentations such as random cropping and flipping to facilitate training.

D. Additional Ablation Studies

D.1. Verification module/model architectures.

In this section, we assess the watermark detection performance of our RAW using diverse model architectures: ResNet 9, ResNet 34, VGG 16 (Simonyan & Zisserman, 2014) and ViT (Dosovitskiy et al., 2020). The summarized results on both Stable-Diffusion-generated MS-COCO and DBDiffusion dataset can be found in Table 5 below. FID and CLIP scores are omitted, given that the watermark strength parameters c_1 and c_2 are consistent with those in the main text (i.e., both set to 0.5). Notably, we observed an improvement in both benign and adversarial detection performances with a more complex model, ResNet 34, which is reasonable as complex models often possess greater learning capacity.

D.2. Size of watermarked training data under fine-tuning scenario.

In this section, we shift our focus to a more realistic scenario where we fine-tune both the classifier and the watermarks using pre-trained models. We note that the new scenario differs from the one discussed in the main text, where the classifier

Table 5: Summary of detection results under different model architectures. AUROC (Ben) denotes the AUROC performance without image manipulations or adversarial attacks. AUROC (Adv) represents the average performance across nine distinct image manipulations and attacks.

Dataset →	MS-COCO		DBDiffusion	
	AUROC (Ben) ↑	AUROC (Adv) ↑	AUROC (Ben) ↑	AUROC (Adv) ↑
ResNet 9	0.94	0.82	0.90	0.80
ResNet 34	0.99	0.93	0.99	0.94
VGG 16	0.99	0.93	0.99	0.90
ViT	0.95	0.88	0.97	0.81

is trained from scratch with randomly initialized weights.

To elaborate, we initially pretrain a set of watermarks along with their corresponding classifiers using a dataset such as MS-COCO. Subsequently, we fine-tune this pair using a new dataset. We present the results of this strategy by varying the number of new training data and summarize the outcomes in Table 6 below. In contrast to the training from scratch scenario described in the main text, our RAW framework demonstrates a significant improvement with a reasonably small training dataset size.

Table 6: Summary of detection results under numbers of training data. AUROC (Ben) denotes the AUROC performance without image manipulations or adversarial attacks. AUROC (Adv) represents the average performance across nine distinct image manipulations and attacks.

Dataset →	MS-COCO		DBDiffusion	
	AUROC (Ben) ↑	AUROC (Adv) ↑	AUROC (Ben) ↑	AUROC (Adv) ↑
$n = 10$	0.74	0.68	0.70	0.68
$n = 50$	0.85	0.71	0.86	0.72
$n = 100$	0.99	0.85	0.99	0.81
$n = 500$	0.99	0.91	0.99	0.93

D.3. Smoothing parameters for training smoothed classifier

In this section, we investigate the watermark detection outcomes of our RAW by varying the smoothing parameters σ, a technique employed to enhance robustness against Gaussian noises. The summarized results for the Stable-Diffusion-generated MS-COCO dataset can be found in Table 7 below. FID and CLIP scores are omitted, as the watermark strength parameters c_1 and c_2 remain consistent with those in the main text (i.e., both set to 0.05).

We observed that as the smoothing parameter σ increases, both benign and adversarial detection performance degrades. This outcome is reasonable, considering that a larger smoothing parameter tends to override all information contained in the original images. In practice, we note that it is important for the user to exercise caution when selecting the smoothing parameter σ, particularly with regard to achieving the desired FPR guarantees when employing our RAW approach.

Table 7: Summary of detection results under different smoothing parameters σ. AUROC (Ben) denotes the AUROC performance without image manipulations or adversarial attacks. AUROC (Adv) represents the average performance across nine distinct image manipulations and attacks.

	AUROC (Ben) ↑	AUROC (Adv) ↑
$\sigma = 0.01$	0.99	0.91
$\sigma = 0.05$	0.98	0.92
$\sigma = 0.1$	0.91	0.87
$\sigma = 0.5$	0.75	0.68

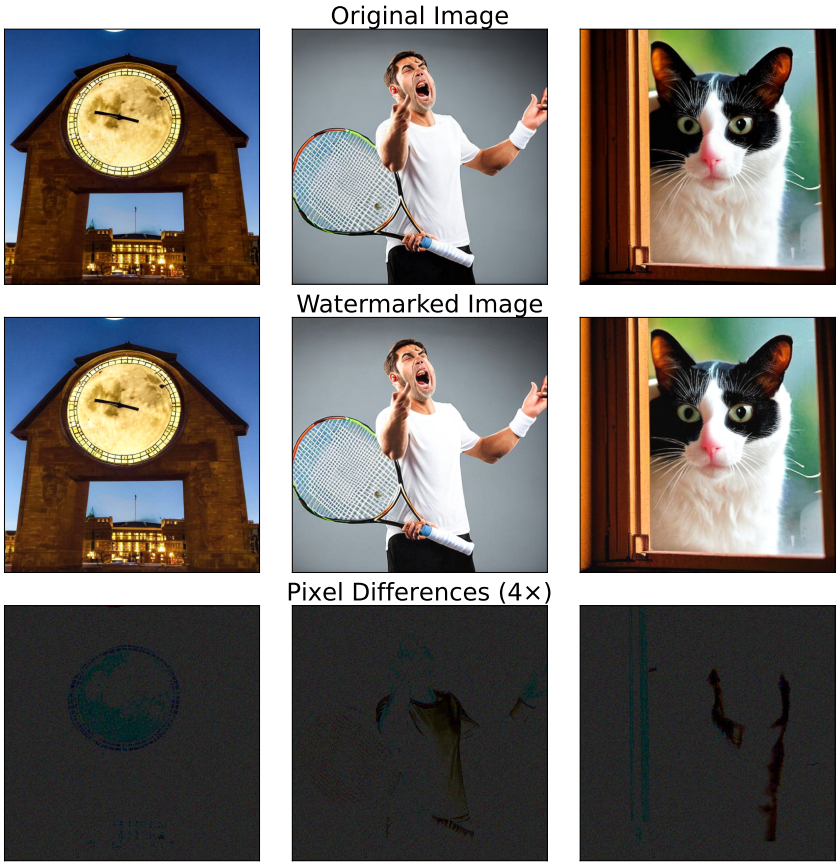

Figure 6: Examples of RAW-watermarked images (middle row)

D.4. Different diffusion models for generating images

In this section, we examine the watermark detection results of our RAW approach across various generative models, specifically utilizing two widely recognized architectures: SDXL and BriXL. We maintain consistent settings as detailed in the main text. The AUROC outcomes are consolidated in Table 8. Our findings indicate that our RAW method achieves high AUROC scores for both MS-COCO and DBDiffusion datasets, underscoring its broad applicability.

Table 8: Summary of detection results under numbers of training data. AUROC (Ben) denotes the AUROC performance without image manipulations or adversarial attacks. AUROC (Adv) represents the average performance across nine distinct image manipulations and attacks.

Dataset →	MS-COCO		DBDiffusion	
	AUROC (Ben) ↑	AUROC (Adv) ↑	AUROC (Ben) ↑	AUROC (Adv) ↑
SDXL	0.98	0.89	0.99	0.87
BriXL	0.99	0.91	0.99	0.89

D.5. Additional Visual Examples

In this section, we present additional visual examples in Figure 6 and 7 resulting from our proposed RAW method. Our observations reveal no discernible differences between the original and watermarked images, thereby emphasizing the efficacy of our approach.

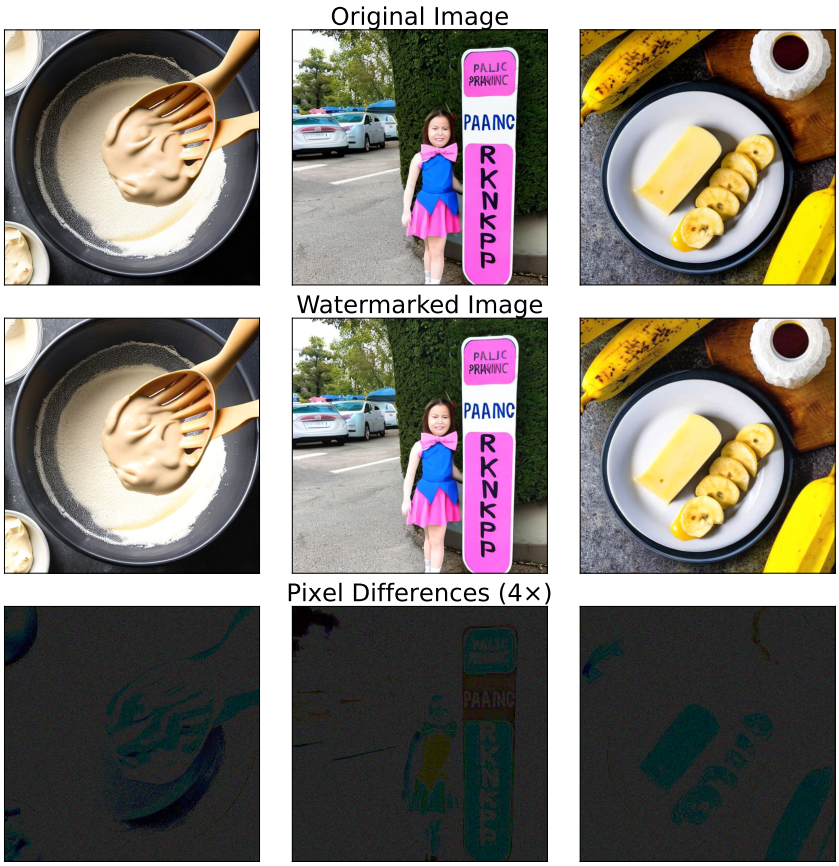

Figure 7: Examples of RAW-watermarked images (middle row)