

---

# Tree-Ring Watermarks: Fingerprints for Diffusion Images that are Invisible and Robust

---

**Yuxin Wen, John Kirchenbauer, Jonas Geiping, Tom Goldstein**  
 University of Maryland

## Abstract

Watermarking the outputs of generative models is a crucial technique for tracing copyright and preventing potential harm from AI-generated content. In this paper, we introduce a novel technique called *Tree-Ring Watermarking* that robustly fingerprints diffusion model outputs. Unlike existing methods that perform post-hoc modifications to images after sampling, *Tree-Ring Watermarking* subtly influences the entire sampling process, resulting in a model fingerprint that is invisible to humans. The watermark embeds a pattern into the initial noise vector used for sampling. These patterns are structured in Fourier space so that they are invariant to convolutions, crops, dilations, flips, and rotations. After image generation, the watermark signal is detected by inverting the diffusion process to retrieve the noise vector, which is then checked for the embedded signal. We demonstrate that this technique can be easily applied to arbitrary diffusion models, including text-conditioned Stable Diffusion, as a plug-in with negligible loss in FID. Our watermark is semantically hidden in the image space and is far more robust than watermarking alternatives that are currently deployed. Code is available at <https://github.com/YuxinWenRick/tree-ring-watermark>.

## 1 Introduction

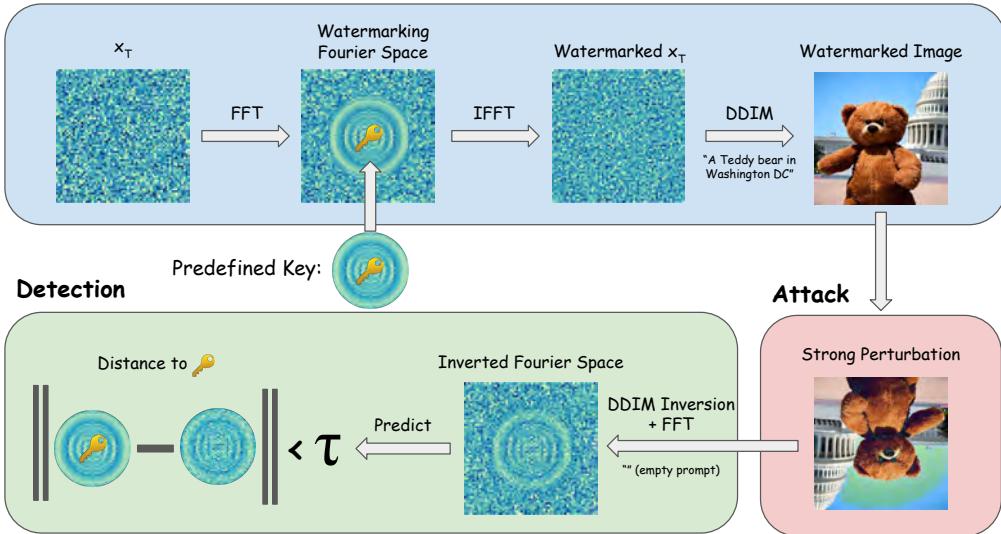
The development of diffusion models has led to a surge in image generation quality. Modern text-to-image diffusion models, like Stable Diffusion and Midjourney, are capable of generating a wide variety of novel images in an innumerable number of styles. These systems are general-purpose image generation tools, able to generate new art just as well as photo-realistic depictions of fake events for malicious purposes.

The potential abuse of text-to-image models motivates the development of *watermarks* for their outputs. A watermarked image is a generated image containing a signal that is invisible to humans and yet marks the image as machine-generated. Watermarks document the use of image generation systems, enabling social media, news organizations, and the diffusion platforms themselves to mitigate harms or cooperate with law enforcement by identifying the origin of an image [Bender et al., 2021, Grinbaum and Adomaitis, 2022].

Research and applications of watermarking for digital content have a long history, with many approaches being considered over the last decade [O’Ruanaidh and Pun, 1997, Langelaar et al., 2000]. However, so far research has always conceptualized the watermark as a minimal modification imprinted onto an existing image [Solachidis and Pitas, 2001, Chang et al., 2005, Liu et al., 2019, Fei et al., 2022]. For example, the watermark currently deployed in Stable Diffusion [Cox et al., 2007], works by modifying a specific Fourier frequency in the generated image.

The watermarking approach we propose in this work is conceptually different: This is the first watermark that is truly invisible, as no post-hoc modifications are made to the image. Instead, the *distribution of generated images is imperceptibly modified and an image is drawn from this*

### Generation



**Figure 1:** Pipeline for *Tree-Ring Watermarking*. A diffusion model generation is watermarked and later detected through ring-patterns in the Fourier space of the initial noise vector.

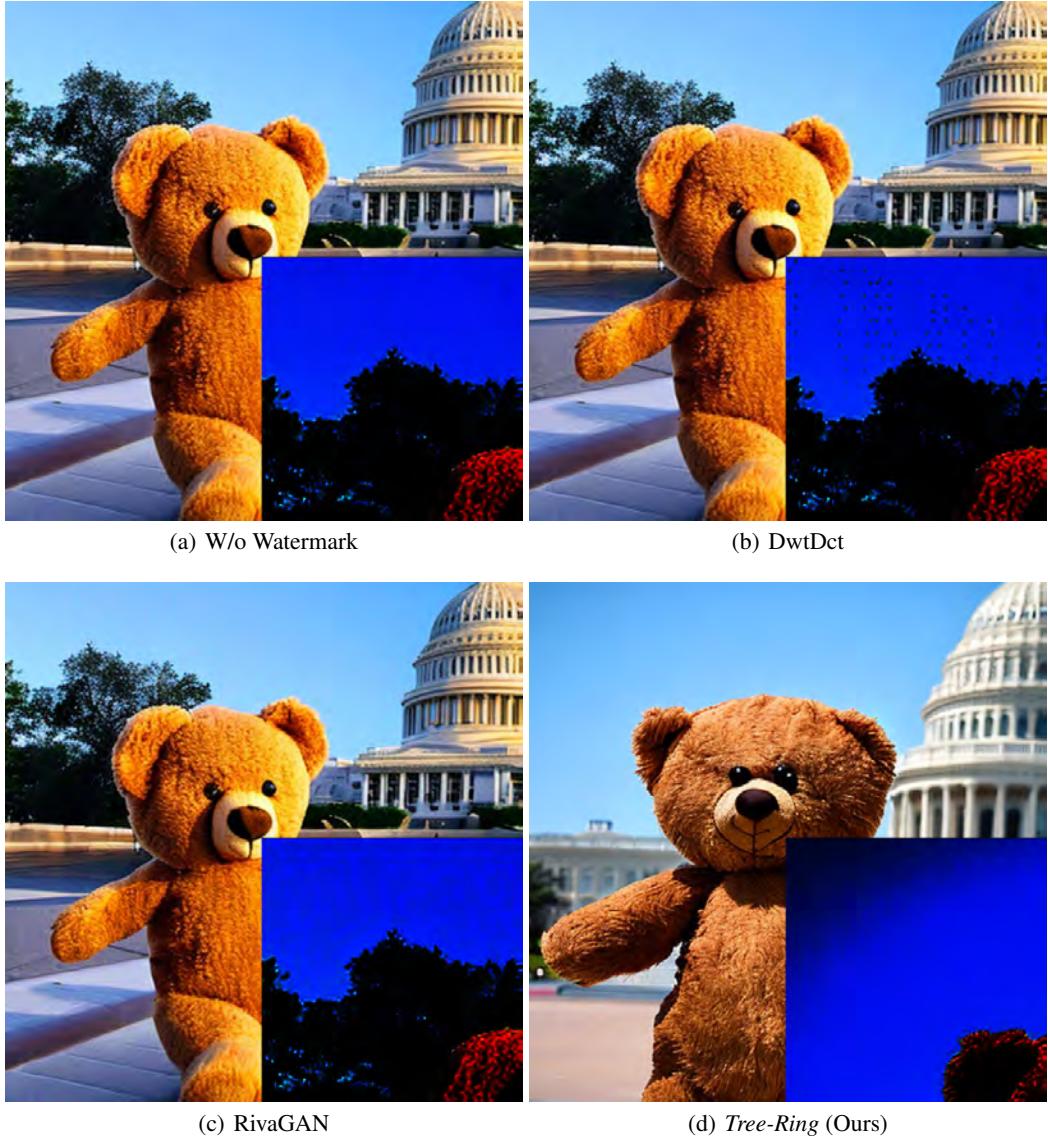
*modified distribution*. This way, the actual sample carries no watermark in the classical additive sense, however an algorithmic analysis of the image can detect the watermark with high accuracy. From a more practical perspective, the watermark materializes in minor changes in the potential layouts of generated scenes, that cannot be distinguished from other random samples by human inspection.

This new approach to watermarking, which we call *Tree-Ring Watermarking* based on the patterns imprinted into the Fourier space of the noise vector of the diffusion model, can be easily incorporated into existing diffusion model APIs and is invisible on a per-sample basis. Most importantly, *Tree-Ring Watermarking* is far more robust than existing methods against a large battery of common image transformations, such as crops, color jitter, dilation, flips, rotations, or noise. *Tree-Ring Watermarking* requires no additional training or finetuning to implement, and the watermark can only be detected by parties in control of the image generation model. We validate the watermark in a number of tests, measuring negligible impact on image quality scores, high robustness to transformations, the low false-positive rate in detection, and usability for arbitrary diffusion models both with and without text conditioning.

## 2 Related Work

**Diffusion Models** Diffusion Models, arising out of the score-based generative models of the formalism of [Song and Ermon \[2019, 2020\]](#), are the currently strongest models for image generation [\[Ho et al., 2020, Dhariwal and Nichol, 2021\]](#). Diffusion models are capable of sampling new images at inference time by iteratively processing an initial noise map. The most prominent sampling algorithm in deployment is DDIM sampling [\[Nichol and Dhariwal, 2021\]](#) without additional noise, which can generate high-quality images in fewer steps than traditional DDPM sampling. Diffusion models are further accelerated for practical usage by optimizing images only in the latent space of a pre-trained VAE, such as in latent diffusion [\[Rombach et al., 2022\]](#).

**Watermarking Digital Content** Strategies to imprint watermarks onto digital content, especially images, have a long tradition in computer vision. Approaches such as [Boland \[1996\]](#), [Cox et al. \[1996\]](#), [O’Ruanaidh and Pun \[1997\]](#) describe traditional watermark casting strategies based on imprinting a watermark in a suitable frequency decomposition of the image, constructed through DCT, DWT, Fourier-Mellin, or complex wavelet transformations. These frequency transformations all share the beneficial property that simple image manipulations, such as translations, rotations, and resizing are easily understandable and watermarks can be constructed with robustness to these transformations in mind. A fair evaluation of watermarking approaches appears in [Pitas \[1998\]](#), [Kutter and Petitcolas \[1999\]](#), which highlight the importance of measurement of false-positive rates for each strategy and ROC-curves under attack through various image manipulations. Work continues



**Figure 2:** Various watermarked generations with the same random seed are presented, showcasing the “invisible” nature of our proposed watermark. A zoomed-in view with high contrast is provided in the bottom right corner. For more high-resolution watermarked images, please refer to Supplementary Material.

on imprinting watermarks, with strategies based on SVD decompositions [Chang et al., 2005], Radon transformations [Seo et al., 2004] and based on multiple decompositions [Al-Haj, 2007].

**Fingerprinting and Watermarking Generative Models** The development of modern deep neural networks opened up new possibilities for “deep” watermarking. Hayes and Danezis [2017] and Zhu et al. [2018] propose strategies to learn watermarking end-to-end, where both the watermark encoder and the watermark decoder are learned models, optimized via adversarial objectives to maximize transmission and robustness [Zhang et al., 2019]. Zeng et al. [2023] present a related approach, in which a neural network watermarked encoder and its associate detector are jointly learned using an image dataset. Notably these approaches still work like a traditional watermark in that the encoder imprints a post-hoc signal onto a given image - however the type of imprint is now learned. We refer to Wan et al. [2022] for an overview. A recent improvement is two-stage processes like Yu et al. [2022], where the trained encoder is used to imprint the watermark onto the training data for a generative model. This leads to a trained generative model where the watermark encoder is

“baked in” to the model, making it easier to generate watermarked data. The Stable Signature of [Fernandez et al. \[2023\]](#), applies this idea to latent diffusion models by finetuning the latent decoder based on a pre-trained watermark encoder. [Zhao et al. \[2023\]](#) similarly train on watermarked data for unconditional diffusion models.

Existing image watermarking approaches first learn a watermark signal and then learn to either embed it into generated data or the generating model. This pipeline stands in contrast to watermarking approaches for language models such as [Kirchenbauer et al. \[2023\]](#). There, no training is necessary to generate watermarked data and the output distribution of the generative model is altered to encode a watermark into generated data in a distributional sense. In the same vein, we propose an approach to alter the output distribution of diffusion models to effectively watermark their outputs. As discussed, this has a number of advantages, in comparison to related work we especially highlight that no training is necessary, that the watermark works with existing models, and that this is the first watermark that does not rely on minor modification of generated images. In this sense, this is the first watermark that is really “invisible”, see [Figure 2](#).

We note in passing that watermarking the output of generative models is not to be confused with the task of watermarking the weights of whole models, such as in [Uchida et al. \[2017\]](#), [Zhang et al. \[2018\]](#), [Bansal et al. \[2022\]](#), who are concerned with identifying and fingerprinting models for intellectual property reasons.

## 2.1 Diffusion Models and Diffusion Inversion

We first introduce the basic notation for diffusion models and DDIM sampling [[Ho et al., 2020](#), [Song and Ermon, 2020](#), [Dhariwal and Nichol, 2021](#)]. A forward diffusion process consists of  $T$  steps of the noise process a predefined amount of Gaussian noise vector to a real data point  $x_0 \in q(x)$ , where  $q(x)$  is the real data distribution, specifically:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t \mathbf{I}), \text{ for } t \in \{0, 1, \dots, T-1\},$$

where  $\beta_t \in (0, 1)$  is the scheduled variance at step  $t$ . The closed-form for this sampling is

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad (1)$$

where,  $\bar{\alpha}_t = \prod_{i=0}^t (1 - \beta_i)$ .

For the reverse diffusion process, DDIM [[Song and Ermon, 2020](#)] is an efficient deterministic sampling strategy, mapping from a Gaussian vector  $x_T \sim \mathcal{N}(0, 1)$  to an image  $x_0 \in q(x)$ . For each denoising step, a learned noise-predictor  $\epsilon_\theta$  estimates the noise  $\epsilon_\theta(x_t)$  added to  $x_0$ . According to [Equation \(1\)](#), we can derive the estimation of  $x_0$  as:

$$\hat{x}_0^t = \frac{x_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(x_t)}{\sqrt{\bar{\alpha}_t}}.$$

Then, we add the estimated noise to  $\hat{x}_0$  to find  $x_{t-1}$ :

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\hat{x}_0^t + \sqrt{1 - \bar{\alpha}_{t-1}}\epsilon_\theta(x_t).$$

We denote such a recursively denoising process from  $x_T$  to  $x_0$  as  $x_0 = \mathcal{D}_\theta(x_T)$ .

However, given the learned model  $\epsilon_\theta(x_t)$ , it is also possible to move in the opposite direction<sup>1</sup>. Starting from an image  $x_0$ , [Dhariwal and Nichol \[2021\]](#) describes an inverse process that retrieves an initial noise vector  $x_T$  which maps to an image  $\hat{x}_0$  close to  $x_0$  through DDIM, where  $\hat{x}_0 = \mathcal{D}_\theta(x_T, 0) \approx x_0$ . This inverse process depends on the assumption that  $x_{t-1} - x_t \approx x_{t+1} - x_t$ . Therefore, from  $x_t \rightarrow x_{t+1}$ , we follow:

$$x_{t+1} = \sqrt{\bar{\alpha}_{t+1}}\hat{x}_0^t + \sqrt{1 - \bar{\alpha}_{t+1}}\epsilon_\theta(x_t).$$

We denote the whole inversion process from a starting real image  $x_0$  to  $x_T$  as  $x_T = \mathcal{D}_\theta^\dagger(x_0)$ .

In this work, we re-purpose DDIM inversion  $\mathcal{D}_\theta^\dagger$  for watermark detection. Given a generated image  $x_0$  with a starting noise  $x_T$ , we apply DDIM inversion to find  $\hat{x}_T$ . We empirically find DDIM’s

<sup>1</sup>To reduce confusion we will always describe the generative diffusion process that goes from  $x_T$  to  $x_0$  as the “reverse process”. We use “inverse process” to denote the estimation of the noise vector  $x_T$  from the final output  $x_0$ .

inversion performance to be quite strong, and  $\hat{x}_T \approx x_T$ . While it may not be surprising that inversion is accurate for unconditional diffusion models, inversion also succeeds well-enough for conditional diffusion models, even when the conditioning  $c$  is not provided. This property of inversion will be exploited heavily by our watermark below.

### 3 Method

In this section, we provide a detailed description of each layer of *Tree-Ring Watermarking*.

#### 3.1 Threat Model

We first briefly describe the threat model considered in this work and clarify the setting: The goal of watermarking is to allow for image generation without quality degradation while enabling the model owner the ability to identify if a given image is generated from their model. Meanwhile, the watermarked image is used in every-day applications and subject to a number of image manipulations and modifications. We formalize this as an adversary who tries to remove the watermark in the generated image to evade detection using common image manipulations, but note that informally, we are also interested in watermark robustness across common usage. Ultimately, this setup leads to a threat model with two agents that act sequentially.

- Model Owner (Generation Phase): Gene owns a generative diffusion model  $\epsilon_\theta$  and allows images  $x$  to be generated through an API containing the private watermarking algorithm  $\mathcal{T}$ . The watermarking algorithm  $\mathcal{T}$  should have a negligible effect on the generated distribution, so that quality is maintained and watermarking leaves no visible trace.
- Forger: Fiona generates an image  $x$  through the API, then tries to evade the detection of  $\mathcal{T}$  by applying strong data augmentations that convert  $x$  to  $x'$ . Later, Fiona uses  $x'$  for a prohibited purpose and claims that  $x'$  is her intellectual property.
- Model Owner (Detection Phase): Given access to  $\epsilon_\theta$  and  $\mathcal{T}$ , Gene tries to determine if  $x'$  originated from  $\epsilon_\theta$ . Gene has no knowledge of the text used to condition the model, or other hyperparameters like guidance strength and the number of generation steps.

#### 3.2 Overview of *Tree-Ring Watermarking*

Diffusion models convert an array of Gaussian noise into a clean image. *Tree-Ring Watermarking* chooses the initial noise array so that its Fourier transform contains a carefully constructed pattern near its center. This pattern is called the “key.” This initial noise vector is then converted into an image using the standard diffusion pipeline with no modifications. To detect the watermark in an image, the diffusion model is inverted using the process described in Section 2.1 to retrieve the original noise array used for generation. This array is then checked to see whether the key is present.

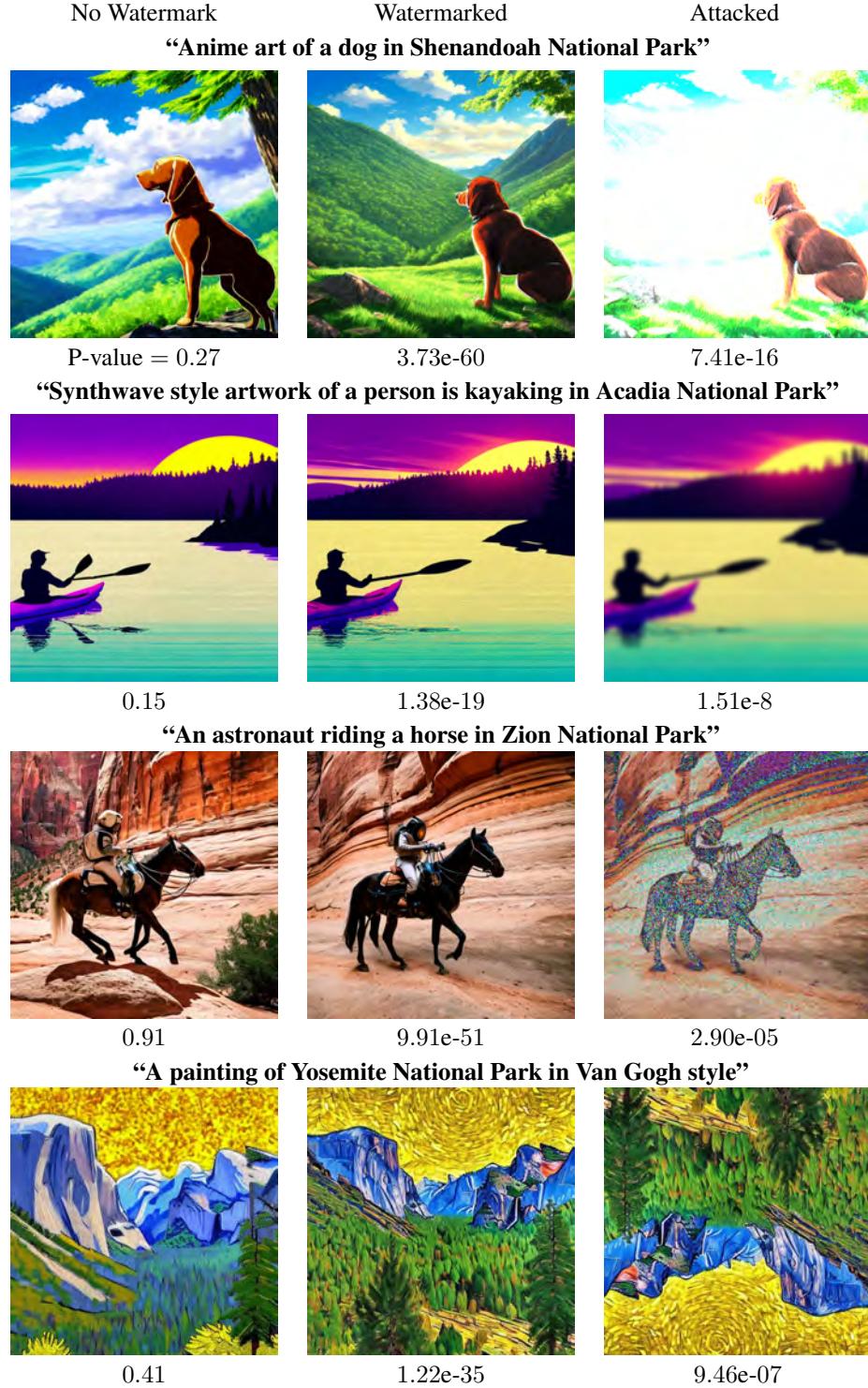
Rather than imprint the key into the Gaussian array directly, which might cause noticeable patterns in the resulting image, we imprint the key into the Fourier transform of the starting noise vector. We choose a binary mask  $M$ , and sample the key  $k^* \in \mathbb{C}^{|M|}$ . As such, the initial noise vector  $x_T \in \mathbb{R}^L$  can be described in Fourier space as

$$\mathcal{F}(x_T)_i \sim \begin{cases} k_i^* & \text{if } i \in M \\ \mathcal{N}(0, 1) & \text{otherwise.} \end{cases} \quad (2)$$

For reasons described below, we choose  $M$  as a circular mask with radius  $r$  centered on the low-frequency modes.

At detection time, given an image  $x'_0$ , the model owner can obtain an approximated initial noise vector  $x'_T$  through the DDIM inversion process:  $x'_T = \mathcal{D}_\theta^\dagger(x'_0)$ . The final metric is calculated as the L1 distance between the inverted noise vector and the key in the Fourier space of the watermarked area  $M$ , i.e.

$$d_{\text{detection distance}} = \frac{1}{|M|} \sum_{i \in M} |k_i^* - \mathcal{F}(x'_T)_i|, \quad (3)$$



**Figure 3:** The qualitative results show three types of images: non-watermarked, *Tree-RingRings* watermarked, and attacked watermarked images. A P-value is provided below each image, which corresponds to the probability of the detected watermark structure occurring by random chance. From top to bottom, the watermarked images are attacked by color jitter with a brightness factor of 6, Gaussian blur with an  $8 \times 8$  filter size, Gaussian noise with  $\sigma = 0.1$ , and a  $180^\circ$  rotation, respectively.

and the watermark is detected if this falls below a tuned threshold  $\tau$ . We later discuss how to calibrate this threshold to a given false-positive either based on a given set of pairs of watermarked and unwatermarked images, or to be set to guarantee a fixed P-value in [Section 3.4](#).

The process described above is straightforward. However, its success depends strongly on the construction of the “key” pattern, which we discuss below.

### 3.3 Constructing a *Tree-Ring* Key

We watermark images by placing a “key” pattern into the Fourier space of the original Gaussian noise array. Our patterns can exploit several classical properties of the Fourier transform for periodic signals that we informally state here.

- A rotation in pixel space corresponds to a rotation in Fourier space.
- A translation in pixel space multiplies all Fourier coefficients by a constant complex number.
- A dilation/compression in pixel space corresponds to a compression/dilation in Fourier space.
- Color jitter in pixel space (adding a constant to all pixels in a channel) corresponds to changing the magnitude of the zero-frequency Fourier mode.

A number of classical watermarking strategies rely on watermarking in Fourier space and exploit similar invariances [[Pitas, 1998](#), [Solachidis and Pitas, 2001](#)]. Our watermark departs from classical methods by applying a Fourier watermark to a random noise array *before* diffusion takes place. Curiously, we will observe below that the invariant properties above are preserved in  $x_T$  even when image manipulations are done in pixel space of  $x_0$ .

In addition to exploiting the invariances above, the chosen key should also be statistically similar to Gaussian noise. Note that the Fourier transform of a Gaussian noise array is also distributed as Gaussian noise. For this reason, choosing a highly non-Gaussian key may cause a distribution shift that impacts the diffusion model.

We consider three different types of keys, with the respective benefits of each pattern being demonstrated in subsequent experimental sections. We believe there are numerous other interesting and practical types that can be explored in future work.

***Tree-Ring*<sub>Zeros</sub>:** We choose the mask to be a circular region to preserve invariance to rotations in image space. The key is chosen to be an array of zeros, which creates invariance to shifts, crops, and dilations. This key is invariant to manipulations, but at the cost of departing severely from the Gaussian distribution. It also prevents multiple keys from being used to distinguish between models.

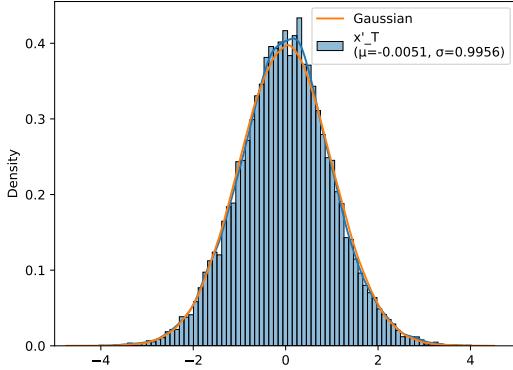
***Tree-Ring*<sub>Rand</sub>:** We draw the a fixed key  $k^*$  from a Gaussian distribution. The key has the same iid Gaussian nature as the original Fourier modes of the noise array, and so we anticipate this strategy will have the least impact on generation quality. This method also offers the flexibility for the model owner to possess multiple keys. However, it is not invariant to make image manipulations.

***Tree-Ring*<sub>Rings</sub>:** We introduce a pattern comprised of *multiple rings*, and constant value along each ring. This makes the watermark invariant to rotations. We choose the constant ring values from a Gaussian distribution. This provides some invariance to multiple types of image transforms, while also ensuring that the overall distribution is only minimally shifted from an isotropic Gaussian.

### 3.4 Deriving P-values for Watermark Detection

A key desideratum for a reliable watermark detector is that it provide an interpretable P-value that communicates to the user how likely it is that the observed watermark could have occurred in a natural image by random chance. In addition to making detection results interpretable, P-values can be used to set the threshold of detection, i.e., the watermark is “detected” when  $p$  is below a chosen threshold  $\alpha$ . By doing so, one can explicitly control the false positive rate  $\alpha$ , making false accusations statistically unlikely.

To this end, we construct a statistical test for the presence of the watermark that produces a rigorous P-value. The forward diffusion process is designed to map images onto Gaussian noise, and so we assume a null hypothesis in which the entries in the array  $x'_T$  obtained for a natural image are Gaussian. We find that this assumption holds quite well in practice, see [Figure 4](#).



**Figure 4:** Histogram of the array  $x'_T$  obtained for a natural image, which is Gaussian.

For any test image  $x'_0$ , we compute the approximate initial vector  $x'_T$  and then set  $y = \mathcal{F}(x'_T)$ . We then define the following null hypothesis

$$H_0 : y \text{ is drawn from a Gaussian distribution } \mathcal{N}(\mathbf{0}, \sigma^2 I_{\mathbb{C}}). \quad (4)$$

Here,  $\sigma^2$  is an unknown variance, which we estimate for each image<sup>2</sup> using the formula  $\sigma^2 = \frac{1}{M} \sum_{i \in M} |y_i|^2$ . To test this hypothesis, we define the score

$$\eta = \frac{1}{\sigma^2} \sum_{i \in M} |k_i^* - y_i|^2. \quad (5)$$

Our formula for  $\eta$  is closely related to [Equation \(3\)](#), but we switch to a sum-of-squares metric and remove the variance from  $y$  to simplify statistical analysis. When  $H_0$  is true, the distribution of  $\eta$  is exactly a *noncentral  $\chi^2$  distribution* [Patnaik, 1949], with  $|M|$  degrees of freedom and non-centrality parameter  $\lambda = \frac{1}{\sigma^2} \sum_i |k_i^*|^2$ .

We declare an image to be watermarked if the value of  $\eta$  is too small to occur by random chance. The probability of observing a value as small as  $\eta$  is given by the cumulative distribution function  $\Phi_{\chi^2}$  of the noncentral  $\chi^2$  distribution:

$$p = \Pr \left( \chi_{|M|, \lambda}^2 \leq \eta \middle| H_0 \right) = \Phi_{\chi^2}(z). \quad (6)$$

$\Phi_{\chi^2}$  is a standard statistical function [Glasserman, 2003], available in `scipy` and many other statistics libraries.

We show qualitative examples of the proposed watermarking scheme and accompanying P-values in [Figure 3](#). For each prompt, we show the generated image with and without the watermark, and also a watermarked image subjected to a transformation. For each image, we report a P-value. As expected, these values are large for non-watermarked images, and small (enabling rejection of the null hypothesis) when the watermark is present. Transformations reduce the watermark strength as reflected in the increased P-value.

## 4 Experiments

We perform experiments on two common diffusion models to measure the efficacy and reliability of the *Tree-Ring Watermarking* technique across diverse attack scenarios. Furthermore, we carry out ablation studies to provide an in-depth exploration of this technique.

### 4.1 Experimental Setting

We employ Stable Diffusion-v2 [Rombach et al., 2022], an open-source, state-of-the-art latent text-to-image diffusion model, along with a  $256 \times 256$  ImageNet diffusion model<sup>3</sup> [Dhariwal and Nichol,

<sup>2</sup>Our statistical test is only sensitive to *Tree-Ring<sub>Rand</sub>* and *Tree-Ring<sub>Rings</sub>*. *Tree-Ring<sub>Zeros</sub>* results in the pathological case that  $\sigma \approx 0$  for watermarked images resulting in overly conservative/large P-values.

<sup>3</sup><https://github.com/openai/guided-diffusion>

**Table 1:** Main Results. T@1%F represents TPR@1%FPR. We evaluate watermark accuracy in both benign and adversarial settings. Adversarial here refers to average performance over a battery of image manipulations. An extended version with additional details and standard error estimates can be found in Supplementary Material.

Model	Method	AUC/T@1%F (Clean)	AUC/T@1%F (Adversarial)	FID ↓	CLIP Score ↑
<b>Stable Diff.</b> FID = 25.29 CLIP Score = 0.363	DwtDct	0.974 / 0.624	0.574 / 0.092	25.10 <sub>.09</sub>	0.362 <sub>.000</sub>
	DwtDctSvd	1.000 / 1.000	0.702 / 0.262	25.01 <sub>.09</sub>	0.359 <sub>.000</sub>
	RivaGAN	0.999 / 0.999	0.854 / 0.448	<b>24.51<sub>.17</sub></b>	0.361 <sub>.000</sub>
	<i>Tree-Ring</i> <sub>Zeros</sub>	0.999 / 0.999	0.963 / <b>0.715</b>	26.56 <sub>.07</sub>	0.356 <sub>.000</sub>
	<i>Tree-Ring</i> <sub>Rand</sub>	1.000 / 1.000	0.918 / 0.702	25.47 <sub>.05</sub>	0.363 <sub>.001</sub>
	<i>Tree-Ring</i> <sub>Rings</sub>	1.000 / 1.000	<b>0.975</b> / 0.694	25.93 <sub>.13</sub>	<b>0.364<sub>.000</sub></b>
<b>ImageNet</b> FID = 17.73	DwtDct	0.899 / 0.244	0.536 / 0.037	17.77 <sub>.01</sub>	-
	DwtDctSvd	1.000 / 1.000	0.713 / 0.187	18.55 <sub>.02</sub>	-
	RivaGAN	1.000 / 1.000	0.882 / 0.509	18.70 <sub>.02</sub>	-
	<i>Tree-Ring</i> <sub>Zeros</sub>	0.999 / 1.000	0.921 / 0.476	18.78 <sub>.00</sub>	-
	<i>Tree-Ring</i> <sub>Rand</sub>	0.999 / 1.000	0.940 / 0.585	18.68 <sub>.09</sub>	-
	<i>Tree-Ring</i> <sub>Rings</sub>	0.999 / 0.999	<b>0.966</b> / <b>0.603</b>	<b>17.68<sub>.16</sub></b>	-

2021]. In the main experiment, we use 50 inference steps for generation and detection for both models. For Stable Diffusion, we use the default guidance scale of 7.5, and we use an empty prompt for DDIM inversion, emulating that the image prompt would be unknown at detection time. The watermark radius  $r$  we use is 10. Later, we conduct more ablation studies on these important hyperparameters. All experiments are conducted on a single NVIDIA RTX A4000.

Our comparative analysis includes three baselines: two training-free methods, DwtDct and DwtDctSvd [Cox et al., 2007], and a pre-trained GAN-based watermarking model, RivaGAN [Zhang et al., 2019, Goodfellow et al., 2014]. However, these baseline methods are designed for steganography, which conceals a target bit-string within an image. To ensure a fair comparison with our exclusively watermarking method, we employ the distance between the decoded bit-string and the target bit-string (Bit Accuracy) as the measurement metric. The approach of Cox et al. [2007] is currently deployed as a watermark mechanism in Stable Diffusion<sup>4</sup>.

## 4.2 Benchmarking Watermark Accuracy and Image Quality

To benchmark the effectiveness of the watermark, we primarily report the area under the curve (**AUC**) of the receiver operating characteristic (ROC) curve, and the True Positive Rate when the False Positive Rate is at 1%, denoted as **TPR@1%FPR**. To demonstrate the generation quality of the watermarked images, we assess the Frechet Inception Distance (**FID**) [Heusel et al., 2017] for both models. Additionally, for the Stable Diffusion model, we also evaluate the **CLIP score** [Radford et al., 2021] between the generated image and the prompt, as measured by OpenCLIP-ViT/G [Cherti et al., 2022]. For AUC and TPR@1%FPR, we create 1,000 watermarked and 1,000 unwatermarked images for each run. For FID, we generate 5,000 images for Stable Diffusion and 10,000 images for the ImageNet Model. The FID of Stable Diffusion is evaluated on the MS-COCO-2017 training dataset [Lin et al., 2014], and the FID of the ImageNet Model is gauged on the ImageNet-1k training dataset [Deng et al., 2009]. All reported metrics are averaged across 5 runs using different random seeds following this protocol.

In Table 1, we present the main experimental results for Stable Diffusion and the ImageNet model. In the clean setting, all baselines except DwtDct and all *Tree-Ring* Watermarking variants are strongly detectable. *Tree-Ring*<sub>Rand</sub> and *Tree-Ring*<sub>Rings</sub> show negligible impact on the FID and no impact on the CLIP score.

<sup>4</sup>[github.com/CompVis/stable-diffusion/blob/main/scripts/txt2img.py#L69](https://github.com/CompVis/stable-diffusion/blob/main/scripts/txt2img.py#L69)

**Table 2:** AUC under each Attack for Stable Diffusion, showing the effectiveness of *Tree-Ring<sub>Rings</sub>* over a number of augmentations. Cr. & Sc. refers to random cropping and rescaling. Additional results for the ImageNet model can be found in Supplementary Material.

Method	Clean	Rotation	JPEG	Cr. & Sc.	Blurring	Noise	Color Jitter	Avg
DwtDct	0.974	0.596	0.492	0.640	0.503	0.293	0.519	0.574
DwtDctSvd	<b>1.000</b>	0.431	0.753	0.511	0.979	0.706	0.517	0.702
RivaGan	0.999	0.173	0.981	<b>0.999</b>	0.974	0.888	0.963	0.854
<b>T-R<sub>Zeros</sub></b>	0.999	<b>0.994</b>	0.984	<b>0.999</b>	0.977	0.877	0.907	0.963
<b>T-R<sub>Rand</sub></b>	<b>1.000</b>	0.486	<b>0.999</b>	0.971	<b>0.999</b>	<b>0.972</b>	<b>0.994</b>	0.918
<b>T-R<sub>Rings</sub></b>	<b>1.000</b>	0.935	<b>0.999</b>	0.961	<b>0.999</b>	0.944	0.983	<b>0.975</b>

### 4.3 Benchmarking Watermark Robustness

To benchmark the robustness of our watermark, we focus on documenting its performance under 6 prevalent data augmentations utilized as attacks. These include  $75^\circ$  rotation, 25% JPEG compression, 75% random cropping and scaling, Gaussian blur with an  $8 \times 8$  filter size, Gaussian noise with  $\sigma = 0.1$ , and color jitter with a brightness factor uniformly sampled between 0 and 6. Additionally, we conduct ablation studies to investigate the impact of varying intensities of these attacks. We report both AUC and TPR@1%FPR in the average case where we average the metrics over the clean setting and all attacks. In all ablation studies, we report the average case.

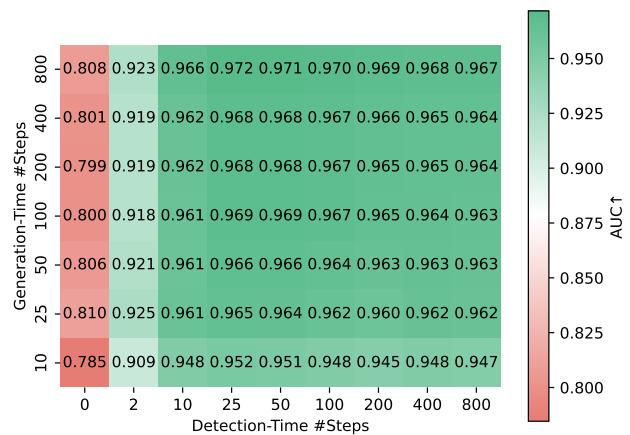
In Table 1, the baseline methods fail in the presence of adversaries. On the contrary, our methods demonstrate higher reliability in adversarial settings. Among them, *Tree-Ring<sub>Rings</sub>* performs the best under adversarial conditions, a result of our careful watermark pattern design.

Further, we show the AUC for each attack setting in Table 2. Notably, *Tree-Ring<sub>Zeros</sub>* demonstrates high robustness against most perturbations, except for Gaussian noise and color jitter. Similarly, *Tree-Ring<sub>Rand</sub>* is robust in most scenarios but performs poorly when faced with rotation, as expected. Overall, *Tree-Ring<sub>Rings</sub>* delivers the best average performance while offering the model owner the flexibility of multiple different random keys. It is worth noting that the baseline method RivaGan also demonstrates strong robustness in most scenarios, but it is important to highlight that our method is training-free and really “invisible”.

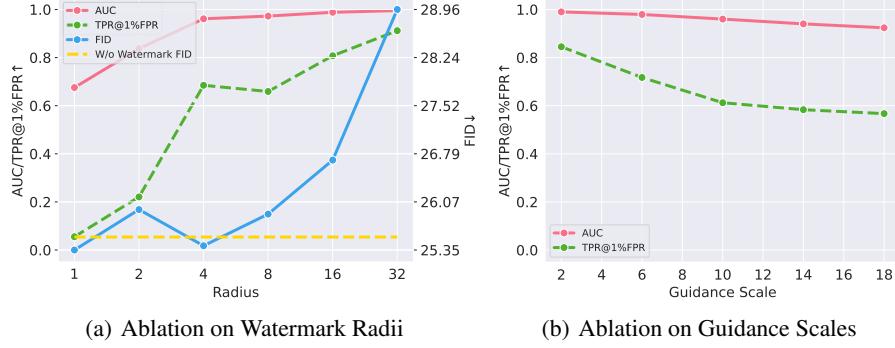
### 4.4 Ablation Experiments

In this section, we undertake exhaustive ablation studies with the Ring pattern on several key hyperparameters to demonstrate the efficacy of *Tree-Ring Watermarking*. Except for the ablation on attacks, the reported numbers represent averages over all attack scenarios and clean images.

In Figure 5, we compare AUC across all step combinations. Surprisingly, even with a significant difference between the generation-time and detection-time #steps, the decrease in AUC is minimal when the model owner uses a reasonable number of inference steps for detection without knowledge of the true generation-time steps. This indicates that the DDIM inversion maintains its robustness in approximating the initial noise vector, and is effective for watermark detection irrespective of the exact number of steps employed. Interestingly, we notice a trend where the detection power appears to be slightly stronger with fewer inference steps at detection time or a larger number of inference steps at generation time. This is an advantageous scenario as the



**Figure 5:** Ablation on Number of Generation Steps versus Detection Steps. Detection succeeds independent of the number of DDIM used to generate data.



**Figure 6:** Ablation on Watermark Radii and Guidance Scales.

model owner now does not actually need to carry out a large number of steps for DDIM inversion, while concurrently, the model owner (or the user) is free to choose the number of generation steps that achieve the best quality [Rombach et al., 2022].

**Number of Steps Used for Generation and Detection.** A key unknown variable for the model owner at the detection time is the actual number of inference steps used during the generation time. This factor could potentially impact the precision of the DDIM inversion approximation of the initial noise vector. To scrutinize this, we systematically vary the number of steps for both the generation and detection time. Due to the computational demands of sampling with a high number of inference steps, we employ a total of 400 images for each run.

**Watermark radii.** The radius of injected watermarking patterns is another critical hyperparameter affecting robustness and generation quality. The corresponding results are shown in Figure 6(a). As the watermarking radius increases, the watermark’s robustness improves. Nevertheless, there is a trade-off with generation quality. We overall confirm a radius of 16 to provide reasonably low FID while maintaining strong detection power.

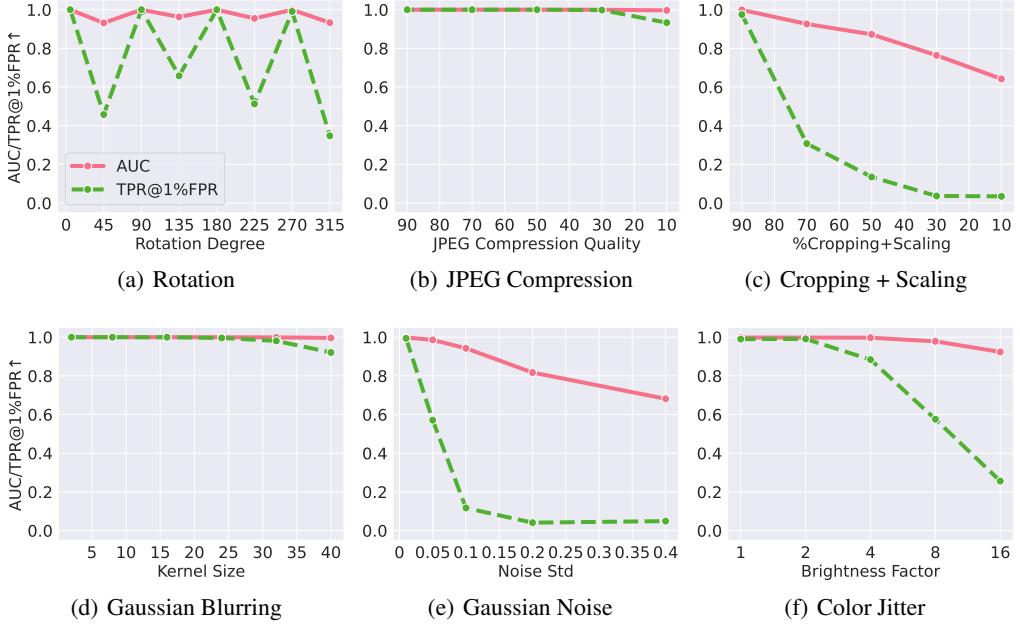
**Guidance scales.** Guidance scale is a hyperparameter that controls the significance of the text condition. Higher guidance scales mean the generation more strictly adheres to the text guidance, whereas lower guidance scales provide the model with greater creative freedom. Optimal guidance scales typically range between 5 and 15 for the Stable Diffusion model we employ. We explore this factor from 2 to 18 in Figure 6(b) and highlight that the strength of the guidance is always unknown during detection time. Although a higher guidance scale does increase the error for DDIM inversion due to the lack of this ground-truth guidance during detection, the watermark remains robust and reliable even at a guidance scale of 18. This is again beneficial for practical purposes, allowing the model owner to keep guidance scale a tunable setting for their users.

**Attack strengths.** Further, we test out the robustness of *Tree-Ring Watermarking* under each attack with various attack strengths. As shown in Figure 7, even with extreme perturbations like Gaussian blurring with kernel size 40, *Tree-Ring Watermarking* can still be reliably detected.

## 5 Limitations and Future Work

*Tree-Ring Watermarking* requires the model owner to use DDIM during inference. Today, DDIM is still likely the most popular sampling method due to its economical use of GPU resources and high quality. However, the proposed watermark will need to be adapted to other sampling schemes should DDIM fall out of favor. Further, the proposed watermark is by design only verifiable by the model owner because model parameters are needed to perform the inversion process. This has advantages against adversaries, who cannot perform a white-box attack on the watermark or even verify whether an ensemble of manipulations broke the watermark. However, it also restricts third parties from detecting the watermark without relying on an API. Finally, it is currently not yet clear how large the capacity for multiple keys  $k^*$  would be. Would it be possible to assign a unique key to every user of the API?

The effectiveness of the proposed watermark is directly related to the accuracy of the inverse DDIM process. Future work that improves the accuracy of this inversion [Zhang et al., 2023], or utilizes



**Figure 7:** Ablation on Different Perturbation Strengths.

invertible diffusion models as described in Wallace et al. [2022], would also improve watermarking power further.

## 6 Conclusion

We propose a new approach to watermarking generative diffusion models using minimal shifts of their output distribution. This leads to watermarks that are truly invisible on a per-sample basis. We describe how to optimally shift, so that the watermark remains detectable even under strong image manipulations that might be encountered in daily usage and handling of generated images.

## 7 Acknowledgements

This work was made possible by the ONR MURI program, DARPA GARD (HR00112020007), the Office of Naval Research (N000142112557), and the AFOSR MURI program. Commercial support was provided by Capital One Bank, the Amazon Research Award program, and Open Philanthropy. Further support was provided by the National Science Foundation (IIS-2212182), and by the NSF TRAILS Institute (2229885).

## References

Ali Al-Haj. Combined DWT-DCT Digital Image Watermarking. *Journal of Computer Science*, 3 (9):740–746, September 2007. ISSN 15493636. doi: 10.3844/jcssp.2007.740.746. URL <http://www.thescipub.com/abstract/?doi=jcssp.2007.740.746>.

Arpit Bansal, Ping-Yeh Chiang, Michael J. Curry, Rajiv Jain, Curtis Wigington, Varun Manjunatha, John P. Dickerson, and Tom Goldstein. Certified Neural Network Watermarks with Randomized Smoothing. In *Proceedings of the 39th International Conference on Machine Learning*, pages 1450–1465. PMLR, June 2022. URL <https://proceedings.mlr.press/v162/bansal22a.html>.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT ’21, pages 610–623, New York, NY, USA, March 2021. Association for Computing Machinery. ISBN 978-1-4503-8309-7. doi: 10.1145/3442188.3445922. URL <https://doi.org/10.1145/3442188.3445922>.

Francis Morgan Boland. Watermarking digital images for copyright protection. 1996. URL <http://www.tara.tcd.ie/handle/2262/19682>.

Chin-Chen Chang, Piyu Tsai, and Chia-Chen Lin. SVD-based digital image watermarking scheme. *Pattern Recognition Letters*, 26(10):1577–1586, July 2005. ISSN 0167-8655. doi: 10.1016/j.patrec.2005.01.004. URL <https://www.sciencedirect.com/science/article/pii/S0167865505000140>.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. *ArXiv*, abs/2212.07143, 2022.

I.J. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for images, audio and video. *Proceedings of 3rd IEEE International Conference on Image Processing*, 3:243–246, 1996. doi: 10.1109/ICIP.1996.560429. URL <http://ieeexplore.ieee.org/document/560429/>.

Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2007. ISBN 9780080555805.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis. *arxiv:2105.05233[cs, stat]*, June 2021. doi: 10.48550/arXiv.2105.05233. URL <http://arxiv.org/abs/2105.05233>.

Jianwei Fei, Zhihua Xia, Benedetta Tondi, and Mauro Barni. Supervised GAN Watermarking for Intellectual Property Protection. *arxiv:2209.03466[cs]*, September 2022. doi: 10.48550/arXiv.2209.03466. URL <http://arxiv.org/abs/2209.03466>.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The Stable Signature: Rooting Watermarks in Latent Diffusion Models. *arxiv:2303.15435[cs]*, March 2023. doi: 10.48550/arXiv.2303.15435. URL <http://arxiv.org/abs/2303.15435>.

Paul Glasserman. *Monte Carlo Methods in Financial Engineering*, volume 53 of *Stochastic Modelling and Applied Probability*. Springer, New York, NY, 2003. ISBN 978-1-4419-1822-2 978-0-387-21617-1. doi: 10.1007/978-0-387-21617-1. URL <http://link.springer.com/10.1007/978-0-387-21617-1>.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL [https://proceedings.neurips.cc/paper\\_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf).

Alexei Grinbaum and Laurynas Adomaitis. The Ethical Need for Watermarks in Machine-Generated Language. *arxiv:2209.03118[cs]*, September 2022. doi: 10.48550/arXiv.2209.03118. URL <http://arxiv.org/abs/2209.03118>.

Jamie Hayes and George Danezis. Generating steganographic images via adversarial training. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL [https://papers.nips.cc/paper\\_files/paper/2017/hash/fe2d010308a6b3799a3d9c728ee74244-Abstract.html](https://papers.nips.cc/paper_files/paper/2017/hash/fe2d010308a6b3799a3d9c728ee74244-Abstract.html).

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html>.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A Watermark for Large Language Models. *arxiv:2301.10226[cs]*, January 2023. doi: 10.48550/arXiv.2301.10226. URL <http://arxiv.org/abs/2301.10226>.

Martin Kutter and Fabien A. P. Petitcolas. Fair benchmark for image watermarking systems. In *Security and Watermarking of Multimedia Contents*, volume 3657, pages 226–239. SPIE, April 1999. doi: 10.1117/12.344672. URL <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/3657/0000/Fair-benchmark-for-image-watermarking-systems/10.1117/12.344672.full>.

G.C. Langelaar, I. Setyawan, and R.L. Lagendijk. Watermarking digital image and video data. A state-of-the-art overview. *IEEE Signal Processing Magazine*, 17(5):20–46, September 2000. ISSN 1558-0792. doi: 10.1109/79.879337.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

Junxiu Liu, Jiadong Huang, Yuling Luo, Lvchen Cao, Su Yang, Duqu Wei, and Ronglong Zhou. An Optimized Image Watermarking Method Based on HD and SVD in DWT Domain. *IEEE Access*, 7:80849–80860, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2915596.

Alex Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. *arxiv:2102.09672[cs, stat]*, February 2021. doi: 10.48550/arXiv.2102.09672. URL <http://arxiv.org/abs/2102.09672>.

J.J.K. O’Ruanaidh and T. Pun. Rotation, scale and translation invariant digital image watermarking. In *Proceedings of International Conference on Image Processing*, volume 1, pages 536–539 vol.1, October 1997. doi: 10.1109/ICIP.1997.647968.

P. B. Patnaik. The Non-Central  $X^2$ - and F-Distribution and their Applications. *Biometrika*, 36(1/2): 202–232, 1949. ISSN 0006-3444. doi: 10.2307/2332542. URL <https://www.jstor.org/stable/2332542>.

I. Pitas. A method for watermark casting on digital image. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(6):775–780, October 1998. ISSN 1558-2205. doi: 10.1109/76.728421.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. URL [https://openaccess.thecvf.com/content/CVPR2022/html/Rombach\\_High-Resolution\\_Image\\_Synthesis\\_With\\_Latent\\_Diffusion\\_Models\\_CVPR\\_2022\\_paper.html](https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html).

Jin S Seo, Jaap Haitsma, Ton Kalker, and Chang D Yoo. A robust image fingerprinting system using the Radon transform. *Signal Processing: Image Communication*, 19(4):325–339, April 2004. ISSN 0923-5965. doi: 10.1016/j.image.2003.12.001. URL <https://www.sciencedirect.com/science/article/pii/S0923596503001541>.

V. Solachidis and L. Pitas. Circularly symmetric watermark embedding in 2-D DFT domain. *IEEE Transactions on Image Processing*, 10(11):1741–1753, November 2001. ISSN 1941-0042. doi: 10.1109/83.967401.

Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. *arXiv:1907.05600 [cs, stat]*, October 2019. URL <http://arxiv.org/abs/1907.05600>.

Yang Song and Stefano Ermon. Improved Techniques for Training Score-Based Generative Models. *arXiv:2006.09011 [cs, stat]*, June 2020. URL <http://arxiv.org/abs/2006.09011>.

Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin’ichi Satoh. Embedding Watermarks into Deep Neural Networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 269–277, Bucharest Romania, June 2017. ACM. ISBN 978-1-4503-4701-3. doi: 10.1145/3078971.3078974. URL <https://dl.acm.org/doi/10.1145/3078971.3078974>.

Bram Wallace, Akash Gokul, and Nikhil Naik. EDICT: Exact Diffusion Inversion via Coupled Transformations. *arxiv:2211.12446[cs]*, December 2022. doi: 10.48550/arXiv.2211.12446. URL <http://arxiv.org/abs/2211.12446>.

Wenbo Wan, Jun Wang, Yunming Zhang, Jing Li, Hui Yu, and Jiande Sun. A comprehensive survey on robust image watermarking. *Neurocomputing*, 488:226–247, June 2022. ISSN 0925-2312. doi: 10.1016/j.neucom.2022.02.083. URL <https://www.sciencedirect.com/science/article/pii/S0925231222002533>.

Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial Fingerprinting for Generative Models: Rooting Deepfake Attribution in Training Data. *arxiv:2007.08457[cs]*, March 2022. doi: 10.48550/arXiv.2007.08457. URL <http://arxiv.org/abs/2007.08457>.

Yu Zeng, Mo Zhou, Yuan Xue, and Vishal M Patel. Securing deep generative models with universal adversarial signature. *arXiv preprint arXiv:2305.16310*, 2023.

Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *ACM Symposium on Information, Computer and Communications Security*. Association for Computing Machinery, Inc., May 2018. ISBN 978-1-4503-5576-6. doi: 10.1145/3196494.3196550. URL <https://research.ibm.com/publications/protecting-intellectual-property-of-deep-neural-networks-with-watermarking>.

Jiaxin Zhang, Kamalika Das, and Sricharan Kumar. On the Robustness of Diffusion Inversion in Image Manipulation. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*, April 2023. URL <https://openreview.net/forum?id=fr8kurMWJIP>.

Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust Invisible Video Watermarking with Attention. *arxiv:1909.01285[cs]*, September 2019. doi: 10.48550/arXiv.1909.01285. URL <http://arxiv.org/abs/1909.01285>.

Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai-Man Cheung, and Min Lin. A Recipe for Watermarking Diffusion Models. *arxiv:2303.10137[cs]*, March 2023. doi: 10.48550/arXiv.2303.10137. URL <http://arxiv.org/abs/2303.10137>.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. HiDDeN: Hiding Data with Deep Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 657–672, 2018. URL [https://openaccess.thecvf.com/content\\_ECCV\\_2018/html/Jiren\\_Zhu\\_HiDDeN\\_Hiding\\_Data\\_ECCV\\_2018\\_paper.html](https://openaccess.thecvf.com/content_ECCV_2018/html/Jiren_Zhu_HiDDeN_Hiding_Data_ECCV_2018_paper.html).

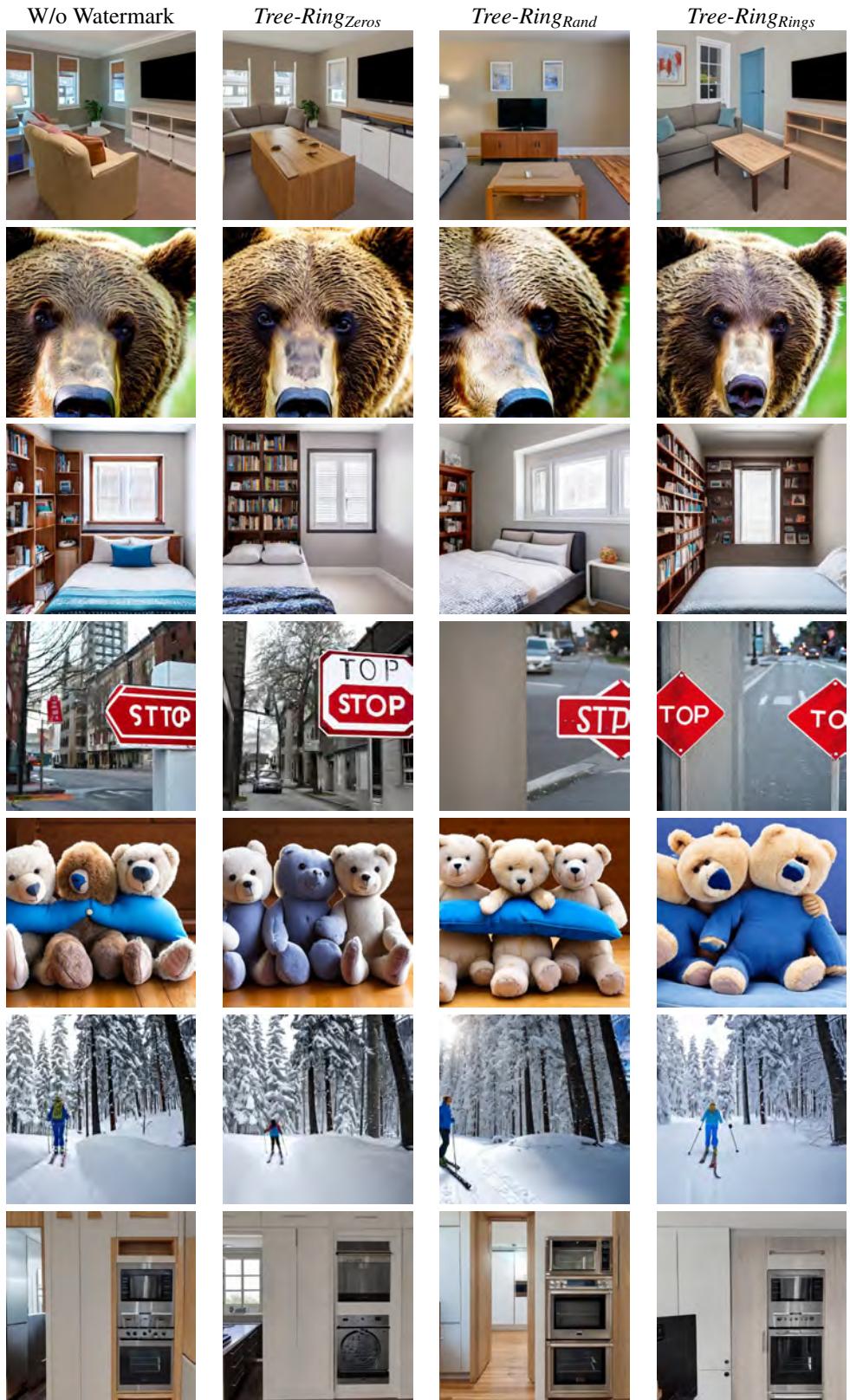
## A Appendix

**Table 3:** Main Results with Error Bars. T@1%F represents TPR@1%FPR. We evaluate watermark accuracy in both benign and adversarial settings. Adversarial here refers to average performance over a battery of image manipulations.

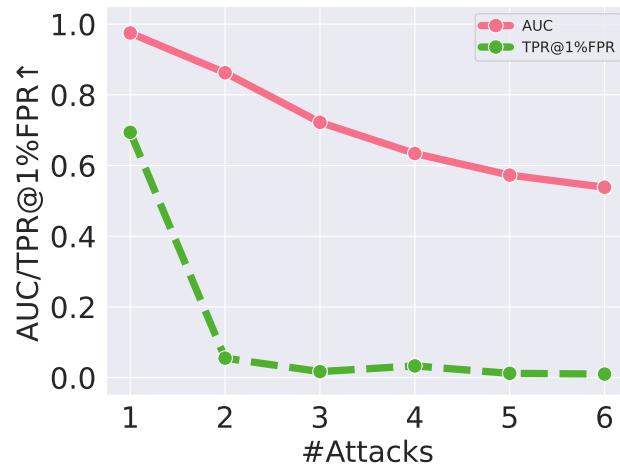
Model	Method	AUC/T@1%F (Clean)	AUC/T@1%F (Adversarial)	FID $\downarrow$	CLIP Score $\uparrow$
<b>Stable Diff.</b> FID = 25.29 CLIP Score = 0.363	DwtDct	0.974 <sub>.001</sub> / 0.624 <sub>.013</sub>	0.574 <sub>.005</sub> / 0.092 <sub>.004</sub>	25.10 <sub>.09</sub>	0.362 <sub>.000</sub>
	DwtDctSvd	1.000 <sub>.000</sub> / 1.000 <sub>.000</sub>	0.702 <sub>.000</sub> / 0.262 <sub>.011</sub>	25.01 <sub>.09</sub>	0.359 <sub>.000</sub>
	RivaGAN	0.999 <sub>.000</sub> / 0.999 <sub>.000</sub>	0.854 <sub>.002</sub> / 0.448 <sub>.006</sub>	<b>24.51<sub>.17</sub></b>	0.361 <sub>.000</sub>
	<b>T-R<sub>Zeros</sub></b>	0.999 <sub>.000</sub> / 0.999 <sub>.000</sub>	0.963 <sub>.001</sub> / <b>0.715<sub>.021</sub></b>	26.56 <sub>.07</sub>	0.356 <sub>.000</sub>
	<b>T-R<sub>Rand</sub></b>	1.000 <sub>.000</sub> / 1.000 <sub>.000</sub>	0.918 <sub>.005</sub> / 0.702 <sub>.017</sub>	25.47 <sub>.05</sub>	0.363 <sub>.001</sub>
	<b>T-R<sub>Rings</sub></b>	1.000 <sub>.000</sub> / 1.000 <sub>.000</sub>	<b>0.975<sub>.001</sub></b> / 0.694 <sub>.018</sub>	25.93 <sub>.13</sub>	<b>0.364<sub>.000</sub></b>
<b>ImageNet</b> FID = 17.73	DwtDct	0.899 <sub>.040</sub> / 0.244 <sub>.203</sub>	0.536 <sub>.016</sub> / 0.037 <sub>.029</sub>	17.77 <sub>.01</sub>	-
	DwtDctSvd	1.000 <sub>.000</sub> / 1.000 <sub>.000</sub>	0.713 <sub>.019</sub> / 0.187 <sub>.008</sub>	18.55 <sub>.02</sub>	-
	RivaGAN	1.000 <sub>.000</sub> / 1.000 <sub>.000</sub>	0.882 <sub>.010</sub> / 0.509 <sub>.009</sub>	18.70 <sub>.02</sub>	-
	<b>T-R<sub>Zeros</sub></b>	0.999 <sub>.000</sub> / 1.000 <sub>.000</sub>	0.921 <sub>.000</sub> / 0.476 <sub>.000</sub>	18.78 <sub>.00</sub>	-
	<b>T-R<sub>Rand</sub></b>	0.999 <sub>.000</sub> / 1.000 <sub>.000</sub>	0.940 <sub>.004</sub> / 0.585 <sub>.006</sub>	18.68 <sub>.09</sub>	-
	<b>T-R<sub>Rings</sub></b>	0.999 <sub>.000</sub> / 0.999 <sub>.000</sub>	<b>0.966<sub>.005</sub></b> / <b>0.603<sub>.006</sub></b>	<b>17.68<sub>.16</sub></b>	-

**Table 4:** AUC under each Attack for the ImageNet model, showing the effectiveness of *Tree-Ring<sub>Rings</sub>* over a number of augmentations. Cr. & Sc. refers to random cropping and rescaling.

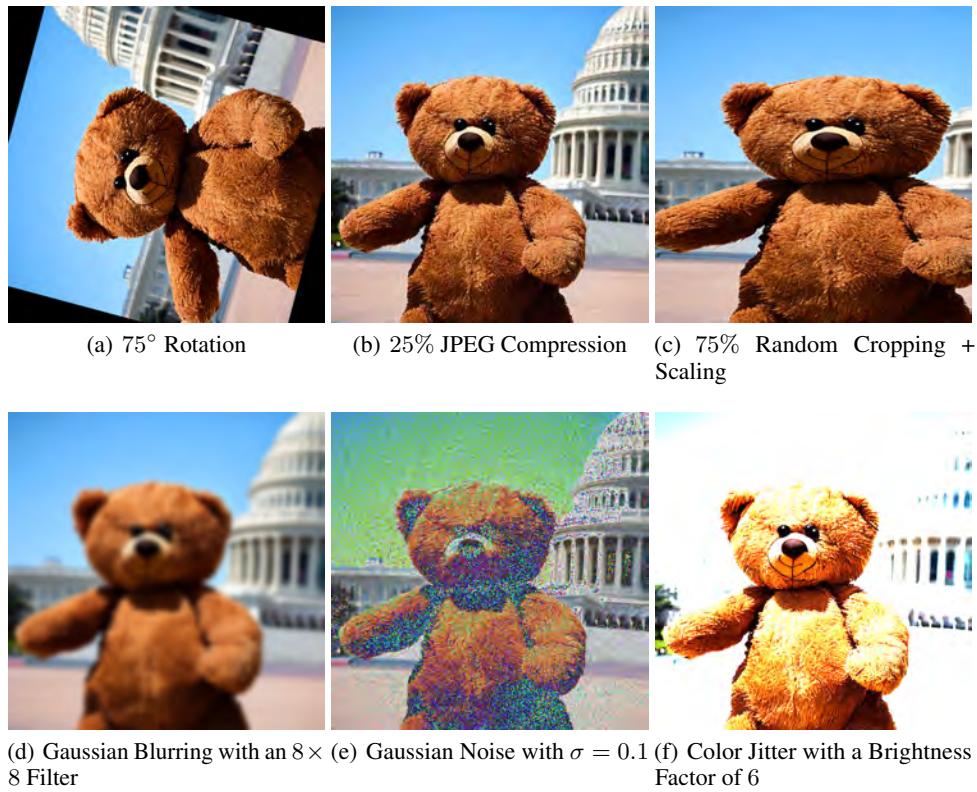
Method	Clean	Rotation	JPEG	Cr. & Sc.	Blurring	Noise	Color Jitter	Avg
DwtDct	0.899	0.478	0.522	0.433	0.512	0.365	0.538	0.536
DwtDctSvd	<b>1.000</b>	0.669	0.568	0.614	0.947	0.656	0.535	0.713
RivaGan	<b>1.000</b>	0.321	<b>0.978</b>	<b>0.999</b>	0.988	0.962	0.924	0.882
<b>T-R<sub>Zeros</sub></b>	0.999	0.953	0.806	0.997	<b>0.999</b>	0.938	0.775	0.921
<b>T-R<sub>Rand</sub></b>	0.999	0.682	0.962	0.997	<b>0.999</b>	<b>0.986</b>	<b>0.956</b>	0.940
<b>T-R<sub>Rings</sub></b>	0.999	<b>0.975</b>	0.940	0.994	<b>0.999</b>	0.979	0.861	<b>0.966</b>



**Figure 8:** More generated images with *Tree-Ring* Watermarking with the first 7 prompts in MS-COCO-2017 training dataset.



**Figure 9:** Results on  $k$  number of random attacks applied at the same time.



**Figure 10:** Attacked images.