RingID: Rethinking Tree-Ring Watermarking for Enhanced Multi-Key Identification

Hai Ci*⬤, Pei Yang*⬤, Yiren Song*⬤, and Mike Zheng Shou✉⬤

Show Lab, National University of Singapore
`cihai03@gmail.com`   `yangpei@u.nus.edu`
`yiren@nus.edu.sg`   `mike.zheng.shou@gmail.com`

Abstract. We revisit *Tree-Ring Watermarking*, a recent diffusion model watermarking method that demonstrates great robustness to various attacks. We conduct an in-depth study on it and reveal that the distribution shift unintentionally introduced by the watermarking process, apart from watermark pattern matching, contributes to its exceptional robustness. Our investigation further exposes inherent flaws in its original design, particularly in its ability to identify multiple distinct keys, where distribution shift offers no assistance. Based on these findings and analysis, we present *RingID* for enhanced multi-key identification. It consists of a novel multi-channel heterogeneous watermarking approach designed to seamlessly amalgamate distinctive advantages from diverse watermarks. Coupled with a series of suggested enhancements, *RingID* exhibits substantial advancements in multi-key identification. Github page: `https://github.com/showlab/RingID`

Keywords: Diffusion Models · Tree-Ring Watermarking · Multi-Key Identification

1 Introduction

With the popularity of diffusion models, a vast number of high-quality diffusion generated images are circulating on the internet. It has become increasingly important to identify AI-generated images to detect fabrication and to track and protect copyrights. Luckily, watermarking [1, 13] offers a reliable solution.

Research on watermarking has a long history in computer vision [9, 34]. A series of works [13, 14, 35, 48, 49] imprint imperceptible watermarks on images through small pixel perturbations. Recently, [47] demonstrates that pixel-level perturbations are provably removable by regeneration attacks. In contrast, *Tree-Ring Watermarking* [38] has recently introduced a novel approach, proposing to imprint a specially designed tree-ring pattern into the initial noise of a diffusion model. Different patterns are also called different "keys". The verification of the watermark's existence involves recovering the initial diffusion noise from the image and comparing the extracted key with the injected one. *Tree-Ring*

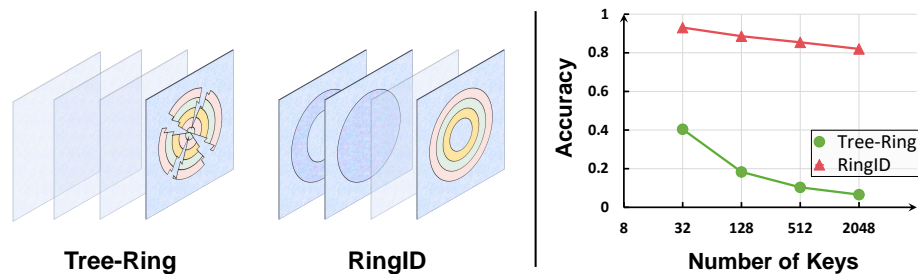* Equal Contribution. ✉Corresponding Author.

Fig. 1: *Tree-Ring* imprints a broken watermark to a single channel. Our method *RingID* achieves stronger robustness by imprinting intact watermarks to multiple channels. In the task of multi-key identification, *RingID* shows overwhelming advantages.

is considered a semantic watermark since the watermark changes the layout of the generated image and semantically hidden in it. It has demonstrated strong robustness against regeneration attacks [47] and various other image transformations [1, 38], credited to the meticulously designed tree-ring pattern. However, previous research only studies its performance in the verification task, which aims at distinguishing between watermarked and unwatermarked images. It's still unclear if *Tree-Ring* can be used to distribute multiple distinct keys and distinguish between them, a.k.a. watermark identification, which is crucial for source tracing and user attribution. Will *Tree-Ring* still be strong and robust as in the verification task? In this work, we will explore and answer this question.

We start by studying the source of *Tree-Ring*'s robustness under verification setting. For the first time, we uncover that the robustness of *Tree-Ring* stems not only from the carefully designed tree-ring pattern but also from the unintentionally introduced distribution shift during watermark imprinting. Such shift makes a big difference, especially in its robustness to Rotation and Crop/Scale image transformations, which cannot be handled by tree-ring's pattern design. More importantly, this overlooked power does not help in the identification scenario, because different watermarks experience the same shift.

We further evaluate *Tree-Ring* in the identification task and surprisingly find that *Tree-Ring* struggles to distinguish between even dozens of different keys. Identification accuracy is reported in Tab. 2. The disappearance of the extra power of distribution shift has clearly exposed more design flaws in *Tree-Ring*. First, we find that tree-ring patterns are not discriminative enough to distinguish between different keys. This results in *Tree-Ring*'s bad accuracy under various attacks or even under the clean setting. Meanwhile, the defective imprinting process leads to the injection of broken watermarks, further weakening its distinguishing ability. In addition, without the help of distribution shift, *Tree-Ring* is completely incapable of handling rotation and crop/scale transformations. Identification accuracy under these attacks is close to zero.

In order to address the above issues, we propose *RingID* for enhanced identification capability. It bases on a novel Multi-Channel Heterogeneous watermarking

framework, which can amalgamate distinctive advantages from different types of watermarks to resist to various kinds of attacks. Coupled with discretization and lossless imprinting methods, *RingID* gains substantial improvement on distinguishing ability. What's more, for the specific attack such as rotation, we locate the flaws in *Tree-Ring*'s original design and propose systematic solutions, which improves the identification accuracy under rotation from nearly 0 to 0.86.

To summarize our contributions:

1. For the first time, we elucidate how the unintentional introduction of distribution shift substantially contributes to *Tree-Ring*'s robustness. We demonstrate this effect from both mathematical and empirical perspectives.
2. We systematically explore *Tree-Ring*'s performance in the identification scenario and reveal its limitations through comprehensive analysis.
3. We present *RingID*, a systematic solution that significantly enhances watermark verification and identification, delivering impressive results. The verification AUC improves from 0.975 to 0.995, while the identification accuracy rises from 0.07 to 0.82.

2 Related Work

2.1 Diffusion Models

Diffusion models [12, 18, 31] signify a swiftly advancing category of generative models. They define a forward diffusion process that gradually add small amount of Gaussian noise to the input data and learn to recover the clean data from the distortion [18]. Diffusion models initially demonstrate huge potential in generating high-resolution images [18, 32], resulting in globally recognized products like Midjourney [25], DALL-E [3, 26] and StableDiffusion [29]. Thereafter, they are broadly applied to modeling various types of data, *e.g.* video [4, 16, 30, 33, 39, 44], audio [19], 3d object [22, 27], humans [7, 8, 36, 50], etc.

2.2 Watermarking Diffusion Models

When it comes to watermarking diffusion models, there are two different types of objectives: watermarking model weights [2, 23, 37, 45, 48] or generated content [6, 14, 35, 43, 46, 49]. Model weights watermarking methods aim to protect the intellectual property of the model itself and identify whether another model is an instance of plagiarism. While content watermarking methods protect the copyright of generated images [13, 42, 43] or videos [24, 46] by adding imperceptible perturbations onto pixels. They either imprint the watermark into the frequency via DFT [20], DCT [9, 11] and wavelet transformations [10, 40], or directly onto pixels via an encoder network [15, 35, 41, 49]. Recently, *Tree-Ring* [38] proposes to imprint a tree-ring watermark pattern into the initial diffusion noise. It demonstrates strong robustness to various attacks. In this work, we delve into *Tree-Ring*, uncovering its source of power in watermark verification and pinpointing its vulnerability in identifying multiple keys. Subsequently, we introduce *RingID*, a systematic solution fortified with enhanced identification capability.

3   Preliminaries

3.1   Threat Model

Model Owner: The owner of the diffusion model provides the image generation service through API access. He would like to imprint imperceptible watermarks into every generated image for copyright protection and source tracking. Given an arbitrary image, the model owner would like to check whether the image contains his own watermark (a.k.a. *Watermark Verification*) and which watermark it is if he distributes multiple different watermarks (a.k.a. *Identification*). The goal of the model owner is to ensure high verification and identification accuracy regardless of any image transformations.

Attacker: The attacker uses the model owner's service to generate an image. He attempts to apply various image transformations such as rotation and JPEG compression to disrupt the watermark, in order to evade the detection and identification check of the model owner. Subsequently, he can claim ownership of the image and profit from it illegally

3.2   Notation

We denote the initial diffusion noise by x_T, and its frequency counterpart after FFT by X_T. T is the diffusion step. Sometimes we may omit the subscript T for simplicity. We denote the real part, imaginary part and conjugate symmetry of X_T by X_{re}, X_{im}, and X_{cs}. In the context of StableDiffusion [29], $X_T \in R^{h \times w \times c}$ has 4 channels $\mathcal{C} = \{0, 1, 2, 3\}$. We denote the watermarked channels by $\mathcal{C}_w$, where $\mathcal{C}_w \subset \mathcal{C}$. We denote the imprinted key by w, the recovered key at detection time by $\hat{w}$, and the null watermark extracted from the unwatermarked image by $\hat{w}_\varnothing$. We use $\mathcal{W} = \{w_1, w_2, \ldots\}$ to denote a set of different keys. $|\mathcal{W}|$ is its capacity.

3.3   Task Formulation: Verification vs. Identification

In this section, we formulate the verification and identification task following the problem statement of *Tree-Ring* [38]. The task of watermark verification is to detect the existence of watermarks. Verification algorithms attempt to distinguish between watermarked images carrying key $\hat{w}$ and unwatermarked images carrying null key $\hat{w}_\varnothing$. More formally, verification can be formulated as distinguishing between the distance to the reference key w: $d(\hat{w}, w) \neq d(\hat{w}_\varnothing, w)$. In *Tree-Ring*, d is chosen to be ℓ_1 distance and w is the watermark to be imprinted.

When a set of different keys $w_i \in \mathcal{W}$ are distributed, the goal of watermark identification is to find the best match for the recovered key $\hat{w}$ among all candidates in $\mathcal{W}$, *i.e.* $\text{ID}(\hat{w}) = \arg\min_i [d(\hat{w}, w_i)]$. In practice, different users are usually assigned different keys. Thus, each generated image can be source tracked and attributed to a unique user via watermark identification.
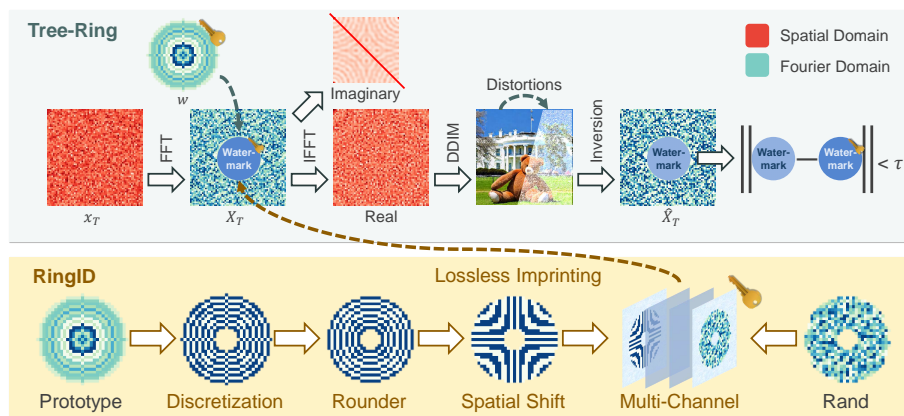
Fig. 2: Framework of the watermarking process. *RingID* introduces a series of approaches that can help to imprint a lossless and robust watermark. In contrast, *Tree-Ring* injects a lossy and less robust watermark.

4 Delve into Tree-Ring Watermarking

In this section, we will reveal why *Tree-Ring* [38] is robust to various image transformations, even some of them are not considered by design. We further evaluate *Tree-Ring* in the multi-key identification task and pinpoint its vulnerabilities.

4.1 Recap of Tree-Ring

Tree-Ring [38] originally studies the verification setting. Its goal is to distinguish watermarked images from non-watermarked images. As illustrated in Fig. 2, *Tree-Ring* transforms the initial diffusion noise x_T into the frequency X_T and injects a tree-ring pattern w into the center of X_T, where the value of each ring in pattern w is sampled from Gaussian. It then transforms the watermarked X_T back to the spatial domain, retaining only the real part to feed the diffusion network for image generation. Notably, the accompanying imaginary part is discarded to be compatible with the network input. For detection of the watermark, *Tree-Ring* inverts the diffusion process to estimate x_T from an image, transforms it to the frequency domain, and extracts the watermark pattern $\hat{w}$. Similarly, nonsense pattern $\hat{w}_\varnothing$ can be extracted from the non-watermarked image. *Tree-Ring* distinguishes between these two ℓ_1-to-reference distances: $\|\hat{w} - w\|_1$ and $\|\hat{w}_\varnothing - w\|_1$ to determine whether an image is watermarked or not.

4.2 Distribution Shift in Watermarking

Tree-Ring's detection algorithm bases on the intuition that the recovered pattern $\hat{w}$ from a watermarked image should match the injected reference pattern w
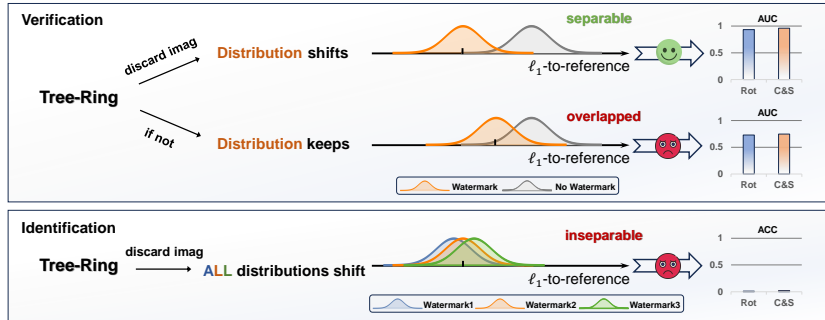
Fig. 3: The effect of distribution shift in both verification and identification tasks. Discarding the imaginary part during watermark imprinting causes the distribution shift in ℓ_1 distance. Distribution shift aids verification but not identification.

to some extent, leading to a smaller ℓ_1-to-reference distance, *i.e.* $\|\hat{w} - w\|_1 < \|\hat{w}_\varnothing - w\|_1$.

However, we have discovered that apart from the pattern matching, another crucial factor contributes to the separation of $\|\hat{w} - w\|_1$ *v.s.* $\|\hat{w}_\varnothing - w\|_1$. This factor is the distribution shift introduced by discarding the imaginary part during the watermark injection process, which is highlighted in Fig. 2. This affects all watermarked images, further reducing the expectation of $\mathbb{E}\left[\|\hat{w} - w\|_1\right]$. In other words, it shifts the distribution of $\|\hat{w} - w\|_1$ towards a smaller value, making it easier to distinguish between $\|\hat{w} - w\|_1$ and $\|\hat{w}_\varnothing - w\|_1$.

Mathematically, we prove that the shift factor is $\frac{\sqrt{3}}{2}$ in a general and simplified setting. Let $\hat{w}_n$ denote the recovered watermark that has never been shifted, thus we have the following:

$$\mathbb{E}\left[\|\hat{w} - w\|_1\right] = \frac{\sqrt{3}}{2}\mathbb{E}\left[\|\hat{w}_n - w\|_1\right]. \tag{1}$$

Detailed derivation is elaborated in Supplementary A. Actually, we find any operation leading to a sufficient distribution shift can be considered a good standalone watermarking approach. We demonstrate the operation of discarding the imaginary part performs well on its own in Supplementary B.

Effect in Different Tasks and Attacks Fig. 3 illustrates the effect of distribution shift under different tasks. In *Verification*, it helps distinguish between watermarked and non-watermarked images. Next, we go one step further and consider the situation when the image undergoes different attacks like JPEG or Resize. If the tree-ring pattern itself can cope with a certain attack, the improvement brought by the distribution shift will be small, because the two distributions are already separable even without the help of shift. On the contrary, if the tree-ring pattern itself cannot deal with a certain attack, the improvement brought by the distribution shift will be significant. Empirically (Supplementary

A.6), we find distribution shift helps a lot under Rotation and Crop/Scale attacks. This implies that the original design of tree-ring cannot handle these two attacks. When it comes to *Identification*, things are a little different. Since different watermarks experience the same distribution shift, discarding the imaginary part doesn't help distinguishing them.

This sparks our curiosity about *Tree-Ring*'s performance in identification, since distribution shift doesn't help and identification is more difficult. We would like to know if *Tree-Ring* still works under Rotation and Crop/Scale attacks. In the next section, we will explore these questions.

4.3   Is Tree-Ring Good at Identification?

We report the performance of *Tree-Ring* when identifying different numbers of keys in Tab. 2. *Tree-Ring* struggles to identify as few as 32 keys. What's more, identification accuracy is nearly zero under Rotation and Crop/Scale attacks.

We summarize 2 reasons as follows: (1) Distribution shift doesn't help. As tree-ring watermark cannot handle Rotation and Crop/Scale attacks, identification accuracy under these attacks drops to nearly zero. (2) Tee-ring pattern doesn't have enough distinguishing ability. Since identification is more difficult than verification, poor distinguishing ability results in poor performance under most attacks or even under no attack setting.

4.4   Why is Tree-Ring Vulnerable to Rotation and Crop/Scale?

All previous experimental results and analysis show one thing, tree-ring pattern cannot handle these two attacks. In this section, we will see what happens.
Vulnerable to Crop & Scale Scaling an image by a factor of a corresponds to compressing the frequency spectrum $X[u,v]$ in both magnitudes and positions: $\frac{1}{a^2}X\left[\frac{u}{a},\frac{v}{a}\right]$. Cropping an image modulates the frequency magnitudes by a sinc function. It is straightforward that these transformations easily disrupt the tree-ring pattern on frequency domain. As shown in Fig. 4(e), the pattern is completely disrupted.
Vulnerable to Rotation Although *Tree-Ring* is designed as ring patterns to seek rotation robustness, we find the existing design ignores many details, leading to no rotation robustness. In short, a combination of corner cropping when rotating, a lossy injection process, and the rotationally asymmetric shape cause the problem. We will explain in detail and propose solutions in Sec. 5.2.

5   RingID: Enhanced Watermarking for Identification

In this section, we show how our proposed method *RingID* advances in identifying multiple distinct keys. *RingID* originates from *Tree-Ring* but incorporates systematic enhancements. We introduce a multi-channel heterogeneous watermarking framework, a discretization approach and a method of lossless imprinting to generally improve its distinguishing ability. We also discuss a series of designs regarding stronger rotation robustness and increased capacity.

5.1   Multi-Channel Heterogeneous Watermarking

Multi-channel heterogeneous (MCH) watermarking serves as a general framework to comprehensively improve watermarking robustness to a battery of attacks. Intuitively, we imprint different types of watermarks on different channels of the initial noise to amalgamate their distinctive advantages, as illustrated in Fig. 1. During matching, we calculate the minimum of channel-wise ℓ_1 distance to the reference watermark for all watermarked channels:

$$\mathrm{ID}\,(\hat{w}) = \arg\min_i \left\{ \min_{c \in C_w}\, [\lambda_c \|\hat{w}^c - w_i^c\|_1] \right\} \tag{2}$$

where c denotes the channel index and traverses all watermarked channels C_w. i denotes the index of candidate reference watermarks. λ_c is a channel-wise normalizing factor.

Eq. (2) bases on the intuition that if a watermark is robust to a certain attack, then its distance from the ground-truth reference is smaller than that of other watermarks that are not robust. Ideally, it enables adaptive selection of the most robust watermark under different attacks. Notably, the total capacity of MCH is the minimum of each watermark. Therefore, it's recommended to use a watermark pattern with sufficient capacity on each channel. At the same time, in order to reduce the impact on the generation quality, we need to carefully select the type of watermarks for combination. We find the Gaussian noise pattern is a perfect candidate. It has infinite capacity and the same distribution as the initial noise. Meanwhile, it has good robustness to non-geometric attacks [38] and is complementary to the tree-ring pattern. So *RingID* explores the combination of a Gaussian noise watermark and a tree-ring watermark. Empirical study in Sec. 6.3 demonstrates that this combination is able to perfectly amalgamate unique advantages from the noise and ring watermark.

5.2   Being Invariant to Rotation

Although the tree-ring pattern is designed to be rotationally symmetric, it exhibits unexpected vulnerability to rotation as shown in Tab. 2. Below, we identify the causes of the issue and propose a series of measures to address it.

Spatial Shift to Avoid Corner Cropping We visualize the imprinted tree-ring watermark in spatial domain in Fig. 4(c). Due to the properties of FFT, tree-ring patterns in the frequency tends to exhibit concentrated energy in the four corners in the spatial domain, rendering them vulnerable to cropping during rotation. We propose to circularly shift the watermark pattern in the spatial domain to the center of the image, thus the majority of watermark information is kept while rotating. For an $N \times N$ latent image, we circularly shift the pattern by $\frac{N}{2}$ pixels in both width and height dimension. This is equivalent to directly multiplying the frequency watermark with a chessboard pattern given by $H[u,v] = e^{j\frac{\pi}{2}(u+v)} = (-1)^{u+v}$. The change of the watermark can be clearly
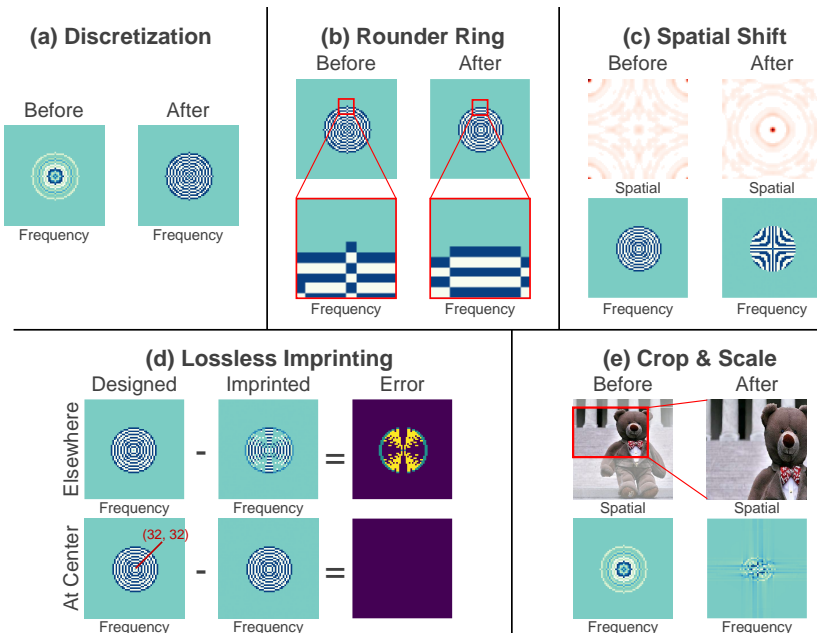
Fig. 4: Detailed illustration of different acts and their impacts. (a) **Discretization**: Discretize values to $\pm\alpha$ for each ring, increasing the distinguishing ability of the watermark. (b) **Rounder ring:** Render smoother edges and reduce artifacts. Increase rotational robustness. (c) **Spatial Shift:** Shift energy on the corners to the center, avoiding potential cropping during rotation. (d) **Lossless Imprinting:** Ensure the imprinted pattern the same as the original design. (e) **Crop & Scale:** Zooming in the spatial domain compresses the imprinted pattern in the frequency domain, causing pattern mismatch and hinders identification.

observed in Fig. 4(c). Although spatial-shifting helps a lot for rotation invariance, we noticed that this often leads to the generation of circular artifacts at the center of the image. Keeping generation quality in mind, we further multiply the shifted pattern by a factor η to suppress the peak at the center. η trades-off the robustness, as smaller η weakens the watermark pattern. In practice, we find $\eta \in [0.8, 0.9]$ strikes a good balance between image quality and robustness.

Lossless Imprinting The operation of discarding the imaginary part during watermarking not only shifts the distribution of ℓ_1-to-reference, but also disrupt the pattern, resulting in the loss of its rotational symmetry. Fig. 4(d) visualizes the actually carried pattern. More concretely, recall from Fig. 2, when the watermarked noise $X[u,v]$ is transformed back to the spatial $x[m,n]$, only its real part $x_{\mathrm{re}}[m,n]$ is kept for image generation. From a frequency domain perspective, this is equivalent to generation with its conjugate symmetry $X_{\mathrm{cs}}[u,v]$, which is not necessarily equal to $X[u,v]$. To leverage the rotational symmetry of the

ring patterns, we attempt to eliminate the information loss during imprinting by solving $X[u,v] = X_{cs}[u,v]$. The results indicate that $X_{re}[u,v]$ should be an even function and $X_{im}[u,v]$ should be an odd function about the Fourier center. To satisfy these constraints, we imprint the tree-ring pattern only in the real part and leave the imaginary part blank. We also align the center of the tree-ring pattern with the Fourier center $(32,32)$. Originally, the tree-ring pattern was put at $(31,32)$ and imprinted on both the real and the imaginary.

Making A Rounder Ring Rings are originally determined by locating pixels equidistant from a give circle center (x_0, y_0): $(x - x_0)^2 + (y - y_0)^2 = r^2$. We observed that this simple approach does not yield a well-rounded circular ring, particularly in images with lower resolutions like initial noise of shape 64×64. Artifacts and alias are shown in Fig. 4(b). Although it is centrosymmetric, it lacks rotational symmetry. Well, the solution is quite straightforward: drawing a rounder ring. There are a number of tools that can help draw a much rounder ring. Here, we employ a very simple idea to get a rounder ring. We place a white pixel on a black background at a distance r from the rotation center. By rotating the low resolution image 360 degrees and recording the trajectory of this pixel, we obtain a rounder ring with a radius of r. Despite its simplicity, empirical results in Tab. 3 show that it is crucial for rotational invariance.

5.3 Discretization for Enhanced Distinguishability

Tree-Ring samples values for each ring from a Gaussian distribution to keep consistency with the distribution of initial noise. We find that this random sampling makes it exceedingly challenging to distinguish between two keys. To address this issue, we propose a discretization approach, specifying that the value for each ring can only be either α or $-\alpha$, illustrated in Fig. 4(a). Consequently, the watermark pattern for n rings has a capacity of 2^n. While this seemingly reduces the upper limit of watermark capacity, it significantly enhances its valid capacity, ensuring optimal utilization of all available slots. Our experiments reveal that setting α as the standard deviation of the initial noise ensures substantial distinguishablity between different keys while reducing the impact on generation quality to a small level.

5.4 Increasing Capacity

We consider two simple approaches to increase capacity: adjusting the number of rings in a single channel or imprinting rings in extra channels. We evaluate the advantages and drawbacks of both methods in Sec. 6.4. In summary, increasing the number of rings $|R|$ in a single channel can effectively boost capacity, leading to $2^{|R|}$ keys. However, excessive ring count results in a noticeable decline in robustness and generation quality. Imprinting watermarks in multiple channels exponentially increases capacity by $n^{|c|}$, where c is number of channels and n is

the capacity of a single channel. However, it's likely to introduce ring-like artifacts in the center of generated images. Therefore, we opt to place the watermark in a single channel and enhance capacity by adjusting the range of rings.

6   Experiment

6.1   Experimental Setup

Following *Tree-Ring* [38], we experiment with the state-of-the-art opensource diffusion model StableDiffusion-V2 [29]. By default, *RingID* imprints a tree-ring watermark with radius 3-14 on the channel 3 of the initial noise and a Gaussian noise watermark on channels 0. We adopt discretization (to 64 and -64), spatial shift, rounder drawing, and lossless imprinting as ordered in Fig. 2. We report the ROC-AUC between 1000 watermarked and 1000 unwatermarked images as the verification metric. In identification, we report identification accuracy with different numbers of keys. Evaluation is performed under various image distortions, following the original setup of *Tree-Ring* [38], *i.e.*, JPEG 25, rotation 75°, 75% random C&S, blur kernel 8, noise std 0.1, brightness [0, 6]. We assess the generation quality with CLIP scores [28] by OpenCLIP-ViT/G [5] between the generated image and the prompts to measure text-image alignment, and Frechet Inception Distance (FID) [17] to quantify the similarity between generated and natural images. we generate 5000 images and evaluate FID on the COCO 2017 training dataset [21]. We set both generation and inversion steps to 50.

6.2   Comparison with Baselines

Robustness In verification (Tab. 1), both *RingID* and *Tree-Ring* perfectly distinguish between watermarked and unwatermarked images when no attacks are performed, achieving AUC score of 1. However, *RingID* exhibit stronger robustness to various image distortions than *Tree-Ring*, achieving near-perfect average AUC score, even though it is not optimized for verification. In the more challenging identification task, (Tab. 2), *RingID* significantly outperforms *Tree-Ring*. When identifying 32 keys, *Tree-Ring*'s average identification accuracy (excluding C&S) is 0.465, while *RingID*'s accuracy reaches up to 0.992. As the number of keys increases to 2048, this gap widens drastically (0.077 *v.s.* 0.942), further highlighting *RingID*'s superior robustness and larger valid key capacity.

Notably, both methods struggle with the C&S distortion in identification (Tab. 2). This aligns with our analysis in Sec. 4.2. Scaling in the spatial domain causes inverse scaling in the frequency domain, leading to pattern disruption and rendering identification challenging under such distortions. However, in verification, with the help of distribution shift, watermarked images can still be effectively distinguished from unwatermarked ones under C&S. Both *Tree-Ring* and *RingID* achieve average AUC higher than 0.97.

Image Quality *RingID* achieves strong performance without compromising the generation quality. It maintains comparable text-image alignments with *Tree-Ring*. Both methods achieve similar CLIP scores (0.364 *v.s.* 0.365). *RingID* obtains an FID of 26.13, which is also comparable to Tree-Ring's FID of 25.93.

Table 1: Comparison with *Tree-Ring* in the verification task. The table shows the ROC-AUC values under various image distortions and CLIP Scores.

Methods	Ring Radius	Clean	Rotate	JPEG	C&S	Blur	Noise	Brightness	Avg	CLIP Score
Tree-Ring	0-10	1.000	0.935	0.999	0.961	0.999	0.944	0.983	0.975	0.364
RingID	0-10	1.000	1.000	1.000	0.979	0.994	0.969	0.991	0.990	0.359
	3-14	1.000	1.000	1.000	0.987	0.989	0.998	0.994	**0.995**	**0.365**

Table 2: Comparison with *Tree-Ring* in the identification task. We report the identification accuracy under various distortions for different numbers of keys.

Methods	#Keys	Clean	Rotate	JPEG	C&S	Blur	Noise	Brightness	Avg	Avg$_{noC\&S}$
Tree-Ring	32	0.790	0.020	0.420	0.040	0.610	0.530	0.420	0.404	0.465
	128	0.450	0.010	0.120	0.020	0.280	0.230	0.170	0.183	0.210
	2048	0.200	0.000	0.040	0.000	0.090	0.070	0.060	0.066	0.077
RingID	32	1.000	1.000	1.000	0.530	0.990	1.000	0.960	**0.926**	**0.992**
	128	1.000	0.980	1.000	0.280	0.980	1.000	0.940	**0.883**	**0.983**
	2048	1.000	0.860	1.000	0.080	0.970	0.950	0.870	**0.819**	**0.942**

6.3   Ablation Study

Components of RingID Tab. 3 ablates all proposed components of *RingID*. We can find that all designs are crucial for the final success of *RingID*. Spatial shift, lossless imprinting, and discretization play a significant role in the robustness of rotation. While the use of rounder rings also increase the accuracy by approximately 40% (from 0.620 to 0.860). Notably, spatial shift sacrifices the robustness to blur and brightness attacks for rotation robustness.

Heterogeneous Watermarking *RingID* employs the combination of a Gaussian noise watermark and a ring watermark. Here, we ablate the choice of imprinting the noise watermark on different channels. By comparing the first 3 rows of Tab. 4, we find that heterogeneous watermarking can effectively combine the advantage from both the noise watermark and the ring watermark. Among single-channel configurations, imprinting noise watermark on channel 0 (0.819) achieves the best performance. The combination of channels 0 and 1 yields the highest accuracy among all trials. Interestingly, imprinting noise watermarks on all available channels (0, 1, 2) decrease the accuracy from 0.830 to 0.807.

6.4   Discussion

Where is the Best Place to Put Rings? We could imprint all rings on one channel or distribute them evenly across multiple channels. In Tab. 5, we fix the total number of rings to 12 and distribute them to 1-4 channels. we can find that performance drops as we distribute rings to more channels. What's more, imprinting rings on multiple channels cause artifacts to appear more frequently (shown in Supplementary C). So given fixed capacity, it's better to imprint all rings on a single channel. This raises another question: what are the optimal

Table 3: Ablation study on the effect of each proposed module in *RingID*. We report the identification accuracy for 2048 keys when removing individual components.

Settings	Clean	Rotate	JPEG	C&S	Blur	Noise	Brightness	Avg
RingID	1.000	0.860	1.000	0.080	0.970	0.950	0.870	**0.819**
- Spatial Shift	1.000	0.000	1.000	0.050	0.990	0.930	0.940	0.701
- Lossless Imprinting	1.000	0.010	0.970	0.160	0.950	0.980	0.830	0.700
- Rounder Ring	1.000	0.620	0.990	0.100	0.890	0.970	0.850	0.774
- Discretization	0.980	0.120	0.380	0.020	0.450	0.650	0.390	0.427
- Heterogeneous	1.000	0.820	0.940	0.030	0.960	0.710	0.720	0.740

Table 4: Ablation study on multi-channel heterogeneous watermarking. This table reports the identification accuracy when the noise watermark is imprinted on different channels (*abbr.* CHs). "-" means the watermark is not imprinted.

Ring CHs	Noise CHs	Clean	Rotate	JPEG	C&S	Blur	Noise	Brightness	Avg
-	0	1.000	0.000	1.000	0.170	0.990	0.950	0.920	0.719
3	-	1.000	0.820	0.940	0.030	0.960	0.710	0.720	0.740
3	0	1.000	0.860	1.000	0.080	0.970	0.950	0.870	<u>0.819</u>
3	1	1.000	0.820	1.000	0.130	1.000	0.880	0.770	0.800
3	2	1.000	0.840	0.980	0.100	0.920	0.990	0.790	0.803
3	0, 1	1.000	0.850	0.950	0.150	0.990	0.990	0.880	**0.830**
3	0, 2	1.000	0.840	0.990	0.120	0.950	0.980	0.870	0.821
3	0, 1, 2	1.000	0.830	0.960	0.160	0.930	0.980	0.790	0.807

inner and outer radii if all rings are placed on a single channel? From Tab. 6 we can see that 3-14 forms a good choice for 11 rings with capacity 2048. The optimal ring radius also varies for different ring numbers.

More Keys We would like to explore whether *RingID* can handle a larger number of keys. Fig. 5 shows the average identification accuracy under various attacks *v.s.* number of keys. We observe that *RingID* maintains good accuracy even with a large number of keys. And identification accuracy linearly decreases as the number of keys exponentially increase.

Table 6: Influence of ring radius on identification accuracy and text-image alignment.

Radius	#Keys	Acc	CLIP Score	Radius	#Keys	Acc	CLIP Score	Radius	#Keys	Acc	CLIP Score
1-6	32	0.886	0.363	1-9	256	0.819	0.362	1-12	2048	0.786	0.361
2-7	32	0.876	0.366	2-10	256	**0.856**	0.358	2-13	2048	0.804	0.361
3-8	32	**0.897**	0.365	3-11	256	0.850	0.362	3-14	2048	**0.819**	**0.368**
4-9	32	**0.897**	**0.368**	4-12	256	0.847	**0.363**	4-15	2048	0.813	0.363
5-10	32	0.881	0.361	5-13	256	0.839	0.360	5-16	2048	0.803	0.360

Table 5: Distributing 12 rings across different number of channels (#CHs). This table reports the average identification accuracy and CLIP Score.

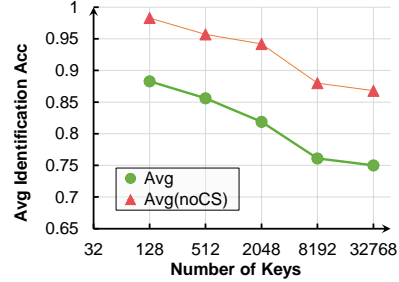| Ring Radius	#CHs	Avg	CLIP Score
3-15	1	**0.740**	**0.357**
3-9	2	0.737	0.351
3-7	3	0.664	0.351
3-6	4	0.544	0.340

Fig. 5: Average identification accuracy *v.s.* number of keys.
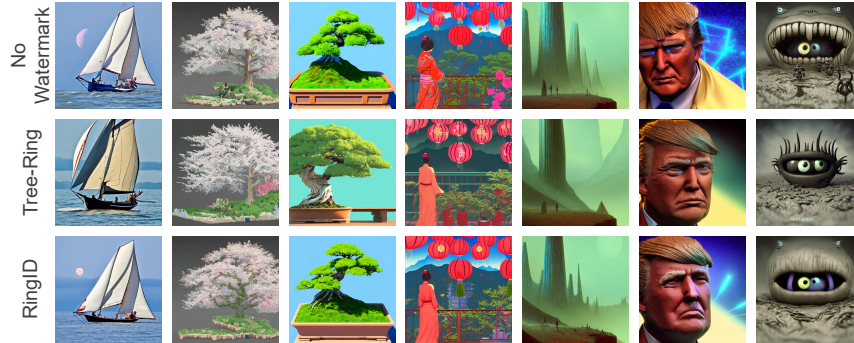
Fig. 6: Qualitative results. The subtle changes in image layouts show that both *Tree-Ring* and *RingID* embed watermarks into the image's semantic content.

6.5   Qualitative Results

As shown in Fig. 6, both *RingID* and *Tree-Ring* watermarking can generate images of good visual quality. Unlike low-level watermarking techniques, both *RingID* and *Tree-Ring* cause subtle changes in the image layout. However, *RingID* generates images with layout that is closer to the unwatermarked image. We owe this to avoiding watermarking on the low-frequency part (radius 0-3).

7   Conclusion and Limitation

We conduct a comprehensive investigation on *Tree-Ring Watermarking*, revealing its limitations in identifying multiple keys. We then propose *RingID* to enhance identification capability.

Limitation: Under multi-key identification, one limitation is that both *Tree-Ring* and our *RingID* are vulnerable to cropping and scaling attacks. Future work may explore other transform domains to resist these attacks.

References

1. An, B., Ding, M., Rabbani, T., Agrawal, A., Xu, Y., Deng, C., Zhu, S., Mohamed, A., Wen, Y., Goldstein, T., et al.: Benchmarking the robustness of image watermarks. arXiv preprint arXiv:2401.08573 (2024)
2. Bansal, A., Chiang, P.y., Curry, M.J., Jain, R., Wigington, C., Manjunatha, V., Dickerson, J.P., Goldstein, T.: Certified neural network watermarks with randomized smoothing. In: International Conference on Machine Learning. pp. 1450–1465. PMLR (2022)
3. Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., et al.: Improving image generation with better captions. Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf **2**(3), 8 (2023)
4. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)
5. Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J.: Reproducible scaling laws for contrastive language-image learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2818–2829 (2023)
6. Ci, H., Song, Y., Yang, P., Xie, J., Shou, M.Z.: Wmadapter: Adding watermark control to latent diffusion models. arXiv preprint arXiv:2406.08337 (2024)
7. Ci, H., Wang, C., Ma, X., Wang, Y.: Optimizing network structure for 3d human pose estimation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2262–2271 (2019)
8. Ci, H., Wu, M., Zhu, W., Ma, X., Dong, H., Zhong, F., Wang, Y.: Gfpose: Learning 3d human pose prior with gradient fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4800–4810 (2023)
9. Cox, I.: Digital watermarking and steganography. Morgan Kaufmann google schola **2**, 893–914 (2007)
10. Cui, S., Guo, X., Zhang, Z.: Estimation and inference in ultrahigh dimensional partially linear single-index models. arXiv preprint arXiv:2404.04471 (2024)
11. Cui, S., Sudjianto, A., Zhang, A., Li, R.: Enhancing robustness of gradient-boosted decision trees through one-hot encoding and regularization. arXiv preprint arXiv:2304.13761 (2023)
12. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)
13. Fernandez, P., Couairon, G., Jégou, H., Douze, M., Furon, T.: The stable signature: Rooting watermarks in latent diffusion models. arXiv preprint arXiv:2303.15435 (2023)
14. Fernandez, P., Sablayrolles, A., Furon, T., Jégou, H., 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)

15. Guo, H., Zhang, Q., Luo, J., Guo, F., Zhang, W., Su, X., Li, M.: Practical deep dispersed watermarking with synchronization and fusion. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 7922–7932 (2023)

16. Guo, Y., Yang, C., Rao, A., Liang, Z., Wang, Y., Qiao, Y., Agrawala, M., Lin, D., Dai, B.: Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. International Conference on Learning Representations (2024)

17. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)

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

19. Kong, Z., Ping, W., Huang, J., Zhao, K., Catanzaro, B.: Diffwave: A versatile diffusion model for audio synthesis. arXiv preprint arXiv:2009.09761 (2020)

20. Lin, C.Y., Wu, M., Bloom, J.A., Cox, I.J., Miller, M.L., Lui, Y.M.: Rotation, scale, and translation resilient watermarking for images. IEEE Transactions on image processing **10**(5), 767–782 (2001)

21. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)

22. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9298–9309 (2023)

23. Liu, Y., Li, Z., Backes, M., Shen, Y., Zhang, Y.: Watermarking diffusion model. arXiv preprint arXiv:2305.12502 (2023)

24. Luo, X., Li, Y., Chang, H., Liu, C., Milanfar, P., Yang, F.: Dvmark: a deep multi-scale framework for video watermarking. IEEE Transactions on Image Processing (2023)

25. Midjourney.com: Midjourney. https://www.midjourney.com/home

26. OpenAI: Dalle2. https://openai.com/dall-e-2

27. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022)

28. 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)

29. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10684–10695 (June 2022)

30. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. arXiv preprint arXiv:2209.14792 (2022)

31. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)

32. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)

33. Song, Y., Huang, S., Yao, C., Ye, X., Ci, H., Liu, J., Zhang, Y., Shou, M.Z.: Processpainter: Learn painting process from sequence data. arXiv preprint arXiv:2406.06062 (2024)
34. Steinberg, Y., Merhav, N.: Identification in the presence of side information with application to watermarking. IEEE Transactions on Information Theory **47**(4), 1410–1422 (2001)
35. Tancik, M., Mildenhall, B., Ng, R.: Stegastamp: Invisible hyperlinks in physical photographs. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2117–2126 (2020)
36. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., Bermano, A.H.: Human motion diffusion model. arXiv preprint arXiv:2209.14916 (2022)
37. Uchida, Y., Nagai, Y., Sakazawa, S., Satoh, S.: Embedding watermarks into deep neural networks. In: Proceedings of the 2017 ACM on international conference on multimedia retrieval. pp. 269–277 (2017)
38. Wen, Y., Kirchenbauer, J., Geiping, J., Goldstein, T.: Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. arXiv preprint arXiv:2305.20030 (2023)
39. Wu, J.Z., Ge, Y., Wang, X., Lei, S.W., Gu, Y., Shi, Y., Hsu, W., Shan, Y., Qie, X., Shou, M.Z.: Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7623–7633 (2023)
40. Xia, X.G., Boncelet, C.G., Arce, G.R.: Wavelet transform based watermark for digital images. Optics Express **3**(12), 497–511 (1998)
41. Xiong, C., Qin, C., Feng, G., Zhang, X.: Flexible and secure watermarking for latent diffusion model. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 1668–1676 (2023)
42. Yang, P., Ci, H., Song, Y., Shou, M.Z.: Steganalysis on digital watermarking: Is your defense truly impervious? arXiv preprint arXiv:2406.09026 (2024)
43. Yu, N., Skripniuk, V., Abdelnabi, S., Fritz, M.: Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In: Proceedings of the IEEE/CVF International conference on computer vision. pp. 14448–14457 (2021)
44. Zhang, D.J., Wu, J.Z., Liu, J.W., Zhao, R., Ran, L., Gu, Y., Gao, D., Shou, M.Z.: Show-1: Marrying pixel and latent diffusion models for text-to-video generation. arXiv preprint arXiv:2309.15818 (2023)
45. Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M.P., Huang, H., Molloy, I.: Protecting intellectual property of deep neural networks with watermarking. In: Proceedings of the 2018 on Asia conference on computer and communications security. pp. 159–172 (2018)
46. Zhang, K.A., Xu, L., Cuesta-Infante, A., Veeramachaneni, K.: Robust invisible video watermarking with attention. arXiv preprint arXiv:1909.01285 (2019)
47. Zhao, X., Zhang, K., Wang, Y.X., Li, L.: Generative autoencoders as watermark attackers: Analyses of vulnerabilities and threats. arXiv preprint arXiv:2306.01953 (2023)
48. Zhao, Y., Pang, T., Du, C., Yang, X., Cheung, N.M., Lin, M.: A recipe for watermarking diffusion models. arXiv preprint arXiv:2303.10137 (2023)
49. Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: Hidden: Hiding data with deep networks. In: Proceedings of the European conference on computer vision (ECCV). pp. 657–672 (2018)
50. Zhu, W., Ma, X., Ro, D., Ci, H., Zhang, J., Shi, J., Gao, F., Tian, Q., Wang, Y.: Human motion generation: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)