METR: Image Watermarking with Large Number of Unique Messages

Alexander Varlamov[1,2], Daria Diatlova[3] and Egor Spirin[2]

[1] *MIPT*
[2] *VK Lab*
[3] *deepvk, VK*

Abstract

Improvements in diffusion models have boosted the quality of image generation, which has led researchers, companies, and creators to focus on improving watermarking algorithms. This provision would make it possible to clearly identify the creators of generative art. The main challenges that modern watermarking algorithms face have to do with their ability to withstand attacks and encrypt many unique messages, such as user IDs. In this paper, we present METR: Message Enhanced Tree-Ring, which is an approach that aims to address these challenges. METR is built on the Tree-Ring watermarking algorithm, a technique that makes it possible to encode multiple distinct messages without compromising attack resilience or image quality. This ensures the suitability of this watermarking algorithm for any Diffusion Model. In order to surpass the limitations on the quantity of encoded messages, we propose METR++, an enhanced version of METR. This approach, while limited to the Latent Diffusion Model architecture, is designed to inject a virtually unlimited number of unique messages. We demonstrate its robustness to attacks and ability to encrypt many unique messages while preserving image quality, which makes METR and METR++ hold great potential for practical applications in real-world settings. Our code is available at https://github.com/deepvk/metr.

Keywords

generative models, diffusion, image watermarking, watermark robustness, message encryption,

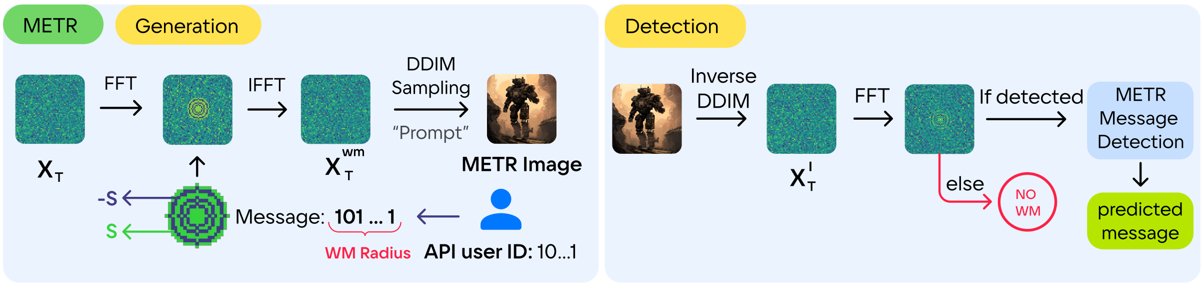
1. Introduction

Nowadays, image generation is one of the major applications of computer vision technology. It is used in various spheres, including entertainment [1, 2], medicine [3], security [4], and retail [5]. Recent advances in deep learning [6], such as Variational Autoencoders (VAE) [7], Generative Adversarial Networks (GAN) [8], and Diffusion models [9], have allowed it to become a rapidly growing area of research. The latest achievements in text-to-image models, namely DALL-E 2 [2], Kandinsky [10], and Stable Diffusion [11], facilitate the generation of highly realistic images based on specific prompts.

With the growing popularity of image generation, the risks of it being used inappropriately or maliciously are also increasing. These risks include policy [1, 12] and privacy violations [13], the generation of fake news [14], document fraud [15, 16], and the creation of harmful content [17]. To help alleviate these risks, it is necessary to develop a mechanism to detect whether an image is generated or not. One possible solution is to label generated images with special messages, watermarks [18, 19, 20]. To ensure the suitability of this solution for practical applications, it is essential that these watermarks remain invisible [21, 20]. Additionally, the watermarks should be robust to a range of attacks [21, 22], ensuring that perturbations to the image cannot remove an encoded message.

Watermarking has been a widely used technique for image content protection long before the emergence of generative models. The first works on the subject introduced algorithms that make it possible to add watermarks to existing images [18, 19, 23]. These methods can be used to apply watermarks to generated images as well. However, this approach has several drawbacks. For example, the watermarks are not necessarily completely invisible to a human's eye [20, 24], and the latest algorithms [25, 26, 27] require training of additional model. Moreover, if the watermarking step is not built into the image generation process, then those with access to the generative model could generate images without watermarks.

CREAI 2024 - Workshop on Artificial Intelligence and Creativity, editors Allegra De Filippo, Francois Pachet, Valentina Presutti, Luc Steels.



(a) METR Generation process

(b) METR Detection process

Figure 1: METR watermarking pipeline. Figure (a) outlines the steps to encrypt a binary message into an image via corresponding latent noise. Figure (b) details the process of detecting whether an image contains a watermark and deciphering the encrypted message.

A different group of algorithms was proposed, which add watermarks to images during the generation process [28, 24, 29]. These approaches successfully address the challenges of previous algorithms. Newer watermarking methods require either tuning of the model's weights [28, 24, 29], or adjustments to some latent image representation (*e.g.,* initial noise), that is used in the generation process [20]. Both approaches make it possible to create "truly invisible" watermarks [20]. Moreover, the second one is applicable to any diffusion architecture with a limitation of only sampling strategy (*e.g.,* DDIM [30]), and it also does not require additional model training. The only disadvantage of the second approach for watermarking images in a generation process can occur when someone gains access to the model's weights. In this scenario, the algorithm responsible for watermark embedding could be removed from the inference pipeline. But this drawback is fixed for the first approach, since the ability to watermark images is contained within the tuned weights of a model. In a recent work by Wen et. al. [20], the authors present *Tree-Ring*, a watermarking algorithm for diffusion models that utilizes initial noise as a latent representation of the image. A watermark is injected as a subtle modification of the initial noise used during diffusion model sampling. This approach shows high robustness to any white-box attacks [21, 22], *i.e.,* attacks only on the output of the model, as the model's weights stay inaccessible.

In addition to being able to determine whether an image is generated or not, it is also important to find out the author of the generated image. This can be done by incorporating watermarks that contain specific information, such as user IDs. Despite the high reliability and imperceptibility of *Tree-Ring* watermarks, the algorithm cannot encrypt messages within the watermark, limiting its practical use for certain real-world scenarios. One of the existing algorithms capable of encrypting messages, *Stable Signature* [24], requires training a separate model for each user, as it can only manage one unique message per model. This is also not ideal for practical use in real-world applications.

In this paper, we propose **METR**, a watermarking algorithm based on *Tree-Ring* watermarking [20] of diffusion models [9, 30]. This approach is able to handle many unique messages and is robust to white-box attacks [21, 22] without a loss in image quality. Similar to *Tree-Ring*, **METR** utilizes a modification of initial noise distribution without changing model architecture. Therefore, it does not require any additional training and can be easily transferred to another model. In addition, we suggest a simple modification of **METR**, **METR++**. It combines **METR** with *Stable Signature* [24] and extends the amount of unique messages encrypted to *as many as one might need.* Our contributions can be summarized as follows:

- **METR** (Message Enhanced *Tree-Ring*) – a new watermarking algorithm that is capable of encoding a large number of unique messages into a *Tree-Ring* watermark without noticeable image quality degradation or a decrease in watermark robustness to attacks.

- We introduce an algorithm to select an optimal value of the hyperparameter S for the **METR** watermark based on the "detection resolution" metric, which captures the difference in detection distances between an image with and without a watermark.

- **METR++** – an extended version of the **METR** watermarking algorithm, combines **METR** with the *Stable Signature* algorithm and allows the encryption of an even larger number of unique messages compared to **METR**.

2. Related Works

2.1. Diffusion Process

Diffusion models [9, 30] are generative models employed to approximate a data distribution $q(\mathbf{x}_0)$ using a parametrized form, $p_\theta(\mathbf{x}_0)$, which is expressed via latent variables $\mathbf{x}_1, ..., \mathbf{x}_T$. The parameters θ are optimized to maximize the evidence lower bound (ELBO).

The forward diffusion process involves the gradual addition of noise to the initial data point $\mathbf{x}_0$: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathrm{I})$, $\bar{\alpha}_t$ are values that parametrize variance at step t for distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1})$.

A point from the initial distribution can be approximated by employing a denoising process. While reverse diffusion was characterized as probabilistic in [9], a deterministic sampling method with an equivalent evidence lower bound (ELBO) was introduced in [30], known as DDIM sampling: $\mathbf{x}_0'^{(t)} := \mathbf{D}_\theta(\mathbf{x}_t) = \dfrac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t} \cdot \epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$, where ϵ_θ is a trained model that predicts the noise added to $\mathbf{x}_0$ to obtain $\mathbf{x}_t$. Meanwhile, $\mathbf{x}_0'^{(t)}$ represents an estimate of $\mathbf{x}_0$ derived from the denoising of $\mathbf{x}_t$.

The determinism of DDIM sampling [30] can be leveraged to trace back the initial noise from which the image was generated. This process is called “inverse diffusion” [20]. We can obtain an estimate of the initial noise, $\mathbf{x}_T'$, based on the assumption that: $\mathbf{x}_{t+1} - \mathbf{x}_t \approx \mathbf{x}_t - \mathbf{x}_{t-1}$. Each step of DDIM inversion mirrors a step of the forward process, but utilizes “trained noise”: $\mathbf{x}_{t+1} = \sqrt{\bar{\alpha}_{t+1}}\mathbf{x}_0'^{(t)} + \sqrt{1-\bar{\alpha}_{t+1}}\epsilon_\theta(\mathbf{x}_t, t)$. T steps of DDIM Inversion estimate the initial noise of an image: $\mathbf{D}_\theta^{inv}(\mathbf{x}_0) = \mathbf{x}_T' \approx \mathbf{x}_T$.

2.2. Tree-Ring Watermark

The *Tree-Ring* watermarking method, proposed in [20], employs the DDIM inversion technique [30]. In this method, the watermark is encoded as concentric circles or squares within the Fourier space of the initial noise: $\mathcal{F}(\mathbf{x}_T)$. This encoding results in a modified version of the initial noise, denoted as $\mathbf{x}_T^{\mathrm{wm}}$. Subsequently, DDIM sampling is applied to produce a watermarked image from the noise that carries the embedded fingerprint: $\mathbf{x}_0^{\mathrm{wm}} = \mathbf{D}_\theta(\mathbf{x}_T^{\mathrm{wm}})$. For watermark detection, the DDIM Inversion process is used to estimate the initial noise and identify the watermark within the Fourier representation of the approximated initial noise of an image: $\mathrm{WM}' = \mathcal{F}(\mathbf{D}_\theta^{inv}(\mathbf{x}_0^{\mathrm{wm}}))$.

The *Tree-Ring* watermark, as presented in [20] transforms the distribution of $\mathbf{x}_T$ into a non-Gaussian form. Since $\mathcal{F}[e^{-ax^2}](q) \sim e^{-\pi^2 q^2/a}$, we can determine whether a watermark is present or absent on the image by assessing if the distribution of $y = \mathcal{F}(\mathbf{x}_T')$, where $\mathbf{x}_T'$ is predicted initial noise in the inverse DDIM process, deviates from normality. Non-normality can be assessed by performing a test on the null hypothesis $\mathcal{H}_0$: $y = \mathcal{F}(\mathbf{x}_T') \sim \mathcal{N}(0, \sigma^2 I)$. σ can be estimated for every input: $\sigma \approx \dfrac{1}{M}\sum_{i=1}^{|M|} |y_i|^2$, where M is a watermarked area of an image. To calculate p-value for $\mathcal{H}_0$, we can define $z(y) = \dfrac{1}{\sigma^2}\sum_{i=1}^{|M|} |\mathrm{WM}_i - y_i|^2$, where WM denotes encrypted watermark values on the area M. We then apply the following equation:

$$p = \mathbb{P}(\chi^2_{|M|,\lambda} < z|\mathcal{H}_0) = F_{\chi^2}(z), \tag{1}$$

where $\chi^2_{|M|,\lambda}$ denotes a non-central chi-squared random variable [31] with $\lambda = \dfrac{1}{\sigma^2}\sum_{i=1}^{|M|} |\mathrm{WM}_i|^2$, and F_{χ^2} representing its cumulative distribution function [32]. Large p-values indicate the presence of a watermark in the image, while small p-values indicate its absence.

2.3. Latent Diffusion and Stable Signature

The Latent Diffusion Model concept, proposed in [11], involves implementing the diffusion process within a latent space, which substantially enhances the quality of image generation. A key architectural update is the incorporation of a Variational Autoencoder (VAE) [7] model, which converts the original image into a compact, low-dimensional representation for use in the subsequent diffusion process. During inference, noise is sampled and then transformed into a latent representation using a trained diffusion model. This representation is finally decoded back into the end image through the decoder component of the VAE.

Drawing on the Latent Diffusion Model concept that leverages VAE, the *Stable Signature* watermarking method was proposed in [24]. *Stable Signature* enables the encryption of binary messages into images during their generation. To embed a *Stable Signature* key into an image, the decoder weights of the Variational Autoencoder (VAE) in the latent diffusion model are fine-tuned. This fine-tuning incorporates a loss function that includes an extra term for message detection: $L = L_{\text{message}} + \lambda L_{\text{image}}$. Message detection is carried out by the pre-trained network W, as detailed in [25]. During the inference process, *Stable Signature* relies solely on the W network to detect and retrieve the watermark. For further information on the Fine-Tuning and Extraction procedures, refer to Figure 2.

The inference process for *Stable Signature* is simple, yet the watermarking algorithm requires a unique Variational Autoencoder (VAE) decoder to be trained for each specific message. In real-world scenarios, especially for services with millions of users where messages are often user IDs, the implementation of *Stable Signature* is not feasible.

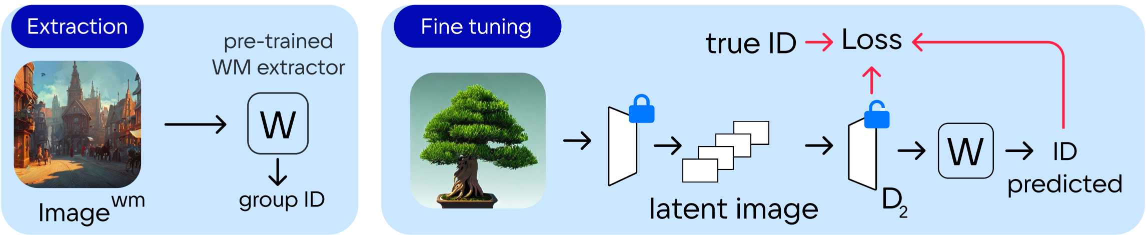
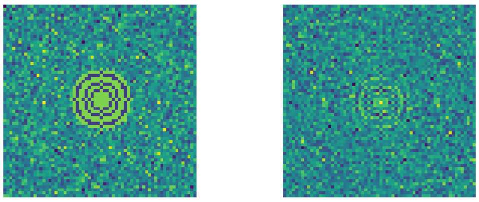


Figure 2: Stable-Signature [24] scheme. Algorithms for watermark extraction and VAE decoder fine-tuning.

2.4. Watermark Attacks

To assess the robustness of watermarking algorithms, it is common practice to not only evaluate detection accuracy metrics, but also to measure the resilience of watermark detection against attacks [25, 20, 24]. In the field of image watermarking, there are standard white-box attacks, as detailed in [33, 21, 22, 21]. These attacks involve transformations of the generated image to verify the robustness of the proposed method, and include operations such as rotation, JPEG compression, and cropping, followed by scaling, Gaussian blur, Gaussian noise, and color jitter. In our study, we also employ attacks utilizing generative models, such as the diffusion model [9, 30] attack described in [21], as well as the VAE [7, 34] attack detailed in [21]. The VAE attack involves embedding an image into the latent space of a Variational Autoencoder (VAE) and then reconstructing it. The diffusion model attack works by denoising injected Gaussian noise on the generated image with the aim of altering the image to erase the watermark.



(a) encrypted message (b) decrypted message

Figure 3: Message becomes circles in Fourier space $\mathcal{F}$ of the latent noise $\mathbf{x}_T$

3. Methods

In this section, we first provide a detailed description of **METR**, the Message Enhanced *Tree-Ring* [20] algorithm and its extension **METR++**. We then present an algorithm designed to select the best hyperparameters for optimal **METR** performance in any use case.

3.1. METR

Figure 1 presents the pipelines for watermark generation and detection using **METR**. Similar to *Tree-Ring* [20], **METR** operates with the noise or latent noise of an image. The watermarking procedure modifies this noise through the corresponding Fourier [35] space. The resulting image can be generated using any diffusion model with the DDIM [30] sampling algorithm.

In **METR**, messages consist of binary sequences encoded using concentric circles with radii increasing from 1 to R, where R is defined as the **watermark radius**. This radius is a fixed hyperparameter representing the number of bits in the message. Therefore, it is possible to encode 2^R messages. Each bit of the message is represented by a single circle, where the ones are assigned the value of S and the zeros are given the value of $-S$. S is another crucial parameter that we call the **message scaler**. Figure 3 illustrates examples of concentric circles with both encrypted and decrypted messages. Algorithm 1 details the pseudocode for generating an image with message encryption using the METR algorithm.

Algorithm 1: METR image generation

```
Input: Scaler $S$, radius $R$, binary message $m$
       (e.g., user ID)
Output: Generated image
$\mathbf{x}_T \sim \mathcal{N}(0, I)$;
$\mathbf{x}_T^{\mathcal{F}} \leftarrow$ Fourier($\mathbf{x}_T$);
for $r = 1$ to $R$ do
    mask$_r \leftarrow$ circle of radius $r$;
    if $m[r] = 1$ then
        $\mathbf{x}_T^{\mathcal{F}}$[mask$_r$] $\leftarrow S$;
    else
        $\mathbf{x}_T^{\mathcal{F}}$[mask$_r$] $\leftarrow -S$;
    end
end
$\mathbf{x}_T^{\text{wm}} \leftarrow$ Inverse Fourier($\mathbf{x}_T^{\mathcal{F}}$);
Image $\leftarrow$ DDIM Sampling($\mathbf{x}_T^{\text{wm}}$);
return Image
```

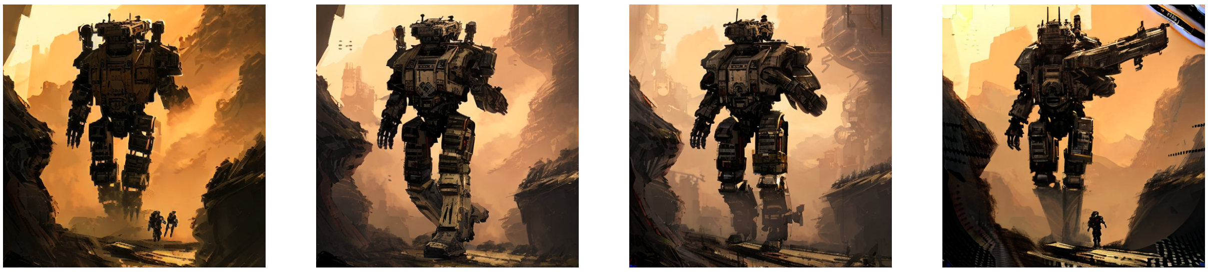
Algorithm 2: METR message detection

```
Input: Image, radius $R$, $p_0$
Output: Predicted watermark
$\mathbf{x}'_T \leftarrow$ DDIM Inversion(Image);
$\mathbf{x}'^{\mathcal{F}}_T \leftarrow$ Fourier($\mathbf{x}'_T$);
if $F_{\chi^2}(z(\mathbf{x}'^{\mathcal{F}}_T)) < p_0$   // see Equation 1
  then
    return NO WM
end
$m \leftarrow []$;
for $r = 1$ to $R$ do
    mask$_r \leftarrow$ circle of radius $r$;
    if $\mathbf{x}'^{\mathcal{F}}_T[mask_r].mean() > 0$ then
        $m.append(1)$;
    else
        $m.append(0)$;
    end
end
return $m$
```

The message can be decoded by reverting the image to its noise using DDIM Inversion, followed by a transformation of the inverted image into the Fourier space. Then, we determine whether the image was watermarked or not by evaluating the p-value using Equation 1 and comparing it with a previously defined threshold p_0.

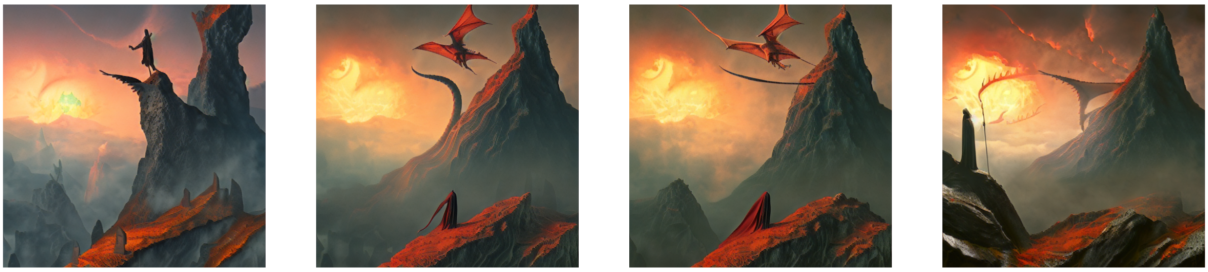
The decryption process for a binary message requires determining the sign of the value for each circle, obtained by averaging all values across the circle.

The corresponding pseudocode for message decryption is shown in Algorithm 2.



(a) no WM (b) $r = 4$ (c) $r = 16$ (d) $r = 34$

Figure 4: Original image and the corresponding one with a **METR** watermark, generated with a fixed scale S, but with different radii r.



(a) no WM (b) $S = 60$ (c) $S = 100$ (d) $S = 140$

Figure 5: Original image and the corresponding one with a **METR** watermark, generated with fixed radius r, but with different scales S.

3.2. Detection Resolution Metric

In this section we describe how to select watermark radius r and message scaler S, which are the parameters of **METR** watermarking algorithm introduced in Section 3.1, and propose the metric Detection Resolution Metric for selecting the scale parameter.

The increase of the radius parameter leads to the increase of the number of the potential messages, and the enlarging scale makes the message more detectable. However, increasing each of the parameter also leads to the production of corrupted images marked by various visible artifacts. See Figure 4 and Figure 5 for examples.

When aiming to encrypt the desired number of messages using the proposed algorithm, it is advisable to estimate the upper bound of possible messages for selecting the radius parameter r. Utilizing the smallest suitable radius is recommended in order to preserve the highest possible image quality.

When choosing the message scaler S, we recommend finding a balance to ensure precise watermark detection and high image quality within the proposed Detection Resolution metric. This metric is designed to capture the contrast between watermarked images and those without watermarks. It is based on the detection distance [20], which is essentially an average error per pixel in the true watermark and restored watermark: $d_{\text{det}}(\mathbf{x}_0) = \frac{1}{|M|} \sum_{i=1}^{|M|} |\text{WM}_i - \text{Detect}(\mathbf{x}_0)_i|$ here M is the watermarked area, WM is a true watermark, $\mathbf{x}_0$ is the original image with a possible watermark and "Detect" function is Fourier transform of DDIM Inversion of its argument.

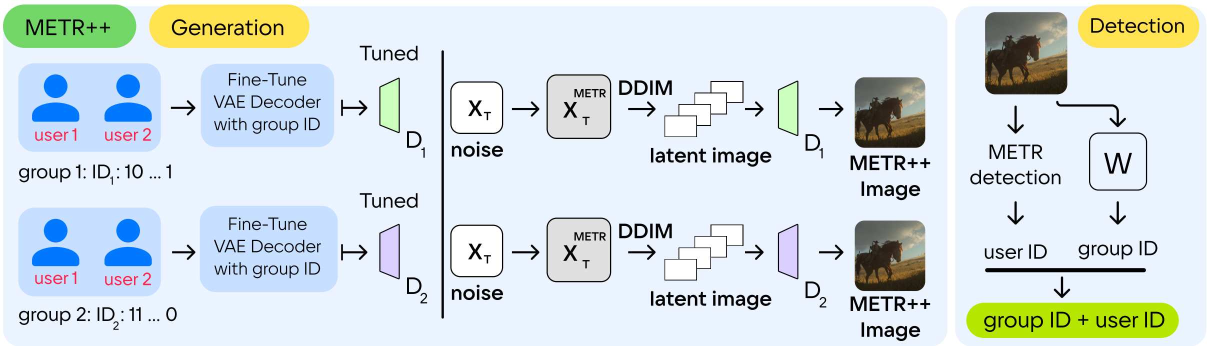


Figure 6: **METR**++ pipeline where messages are divided into groups. **METR** is used to encode messages inside a group, and *Stable Signature* [24] is used to encode group itself.

Detection Resolution metric computes the difference in detection distances between an image without a watermark $\mathbf{x}_0$ and an image with one $\mathbf{x}_0^*$:

$$R_{\text{det}}(\mathbf{x}_0, \mathbf{x}_0^*) = d(\mathbf{x}_0) - d(\mathbf{x}_0^*) \tag{2}$$

A binary message m is considered accurately detected if the inverse process error does not negate the value of S. This means that $S - d(\mathbf{x}_0^*) < 0$ should hold for watermarked images, and $S - d(\mathbf{x}_0) > 0$ for non-watermarked ones. Furthermore, it is essential that the detection resolution metric, which is the error contrast between watermarked and non-watermarked images, is a relatively large value, compared to S. To specify this, we computed $g = R_{\text{det}}/S$ for situations with a perfect detection of message, and it occurred that $g = kS + b$, where k and b are linear fit parameters. It means that to prevent detection collisions, we need $g \geq kS + b$ or $R_{\text{det}}/(kS^2 + bS) \geq 1$. During early experiments, we found out that the most reliable k and b are $-2.23 \cdot 10^{-3}$ and 0.653 respectively.

Since the error of the DDIM Inversion [30] process with a fixed model remains relatively constant with respect to the chosen image, we can assess the detection accuracy for a given value of S using just a single pair of $(\mathbf{x}_0, \mathbf{x}_0^*)$. This way, we bring the selection of S down to defining the possible range of its value and selecting the best one based on a subset of images. The algorithm consists of two steps. Initially, it measures the detection resolution using a generated image with and without watermark from the same noise. Then, if the criterion is satisfied, it assesses the image quality across the full test dataset. See Algorithm 3 for details.

Algorithm 3: Parameter S search

```
Input: latent noise $\mathbf{x}_T \sim \mathcal{N}(0, I)$, test dataset $\mathbf{D} = \langle \text{prompt}, \mathbf{x}_{\text{true}} \rangle$, image quality threshold $\Theta$,
       $k$, $b$, radius $r$
Output: best message scaller $S$
$\mathbf{x}_T^{\mathcal{F}} \leftarrow \text{Fourier}(\mathbf{x}'_T)$;
$p_{\text{min}}, p_{\text{max}} \leftarrow \min(\mathbf{x}_T^{\mathcal{F}}), \max(\mathbf{x}_T^{\mathcal{F}})$;
for $S = p_{min}$ to $S = p_{max}$; step do
    $\mathbf{x}_0 \leftarrow \text{Generate}(\mathbf{x}_T)$;
    $\mathbf{x}_0^* \leftarrow \text{Generate}(\text{METR}(\mathbf{x}_T, S, r))$;
    $d_0, d_0^* \leftarrow d(x_0), d_(x_0^*)$;
    Det_Res $\leftarrow \frac{R_{\text{det}}(d_0, d_0^*)}{kS^2 + bS}$;
    if $d_0 > S$ and $d_0^* < S$ and Det_Res $\geq 1$ then
        $D_{\text{test}} \leftarrow$ Generate from prompts in $D$ with WM;
        $L \leftarrow \text{Eval}(D_{\text{test}}, D)$                  // i.e., FID;
        if $L < \Theta$ then
            Return S
        end
    end
end
```

3.3. METR++

Although **METR** already provides support for encoding messages into images, the selected radius r still restricts their number. To overcome this, we developed **METR++**, an extension of the original **METR** algorithm. **METR++** is designed to significantly increase the original algorithm's potential for encoding various unique messages, while being limited in terms of models that it can be applied to. The extended algorithm can only be used with Latent Diffusion [11], which is currently the most commonly used image generation architecture [11, 2, 10]. **METR++** incorporates two watermarks into an image. As demonstrated in Figure 6, it adopts the **METR** watermarking algorithm 3.1, augmented by a VAE decoder derived from the latent diffusion model [11], as proposed in *Stable Signature* [24].

To encode multiple messages, they are first categorized into groups with a capacity of 2^r, a size that reflects the number of potential unique messages in the **METR** watermarking algorithm. Then, each group is assigned a distinct ID, and uses a specifically fine-tuned VAE decoder designed to encode the group's ID as a *Stable Signature* watermark. Consequently, **METR++** expands the capacity for encoding unique messages within **METR**, multiplying it by the number of specially fine-tuned VAE decoders for each group.

The identification process consists of two parts, that can be done in parallel. First is the group's unique key decryption via the *Stable Signature* approach. Second is the **METR** message decoding, which identifies a user within a group. Decrypting the **METR** message requires carrying out the inverse diffusion process on the image latent, that is obtained with a VAE encoder part of LDM, which weights stay intact.

To conclude, **METR++** can be adapted to any Latent Diffusion model and can encrypt approximately $2^r \cdot n$ unique messages, where r is **METR**'s watermark radius, and n is the number of fine-tuned VAE decoders.

4. Experiments

In this section, we describe our experimental setup and present the results of experiments conducted using **METR** and **METR++**.

4.1. Experimental Setup

In all experiments, we utilize the base version of Stable Diffusion 2.1 [11] and limit the generation process to 40 steps of DDIM [30] sampling. The inverse diffusion process used for **METR** message detection was run with no prompt for the same number of steps as the generation process. The guidance scale was set to 7.5.

For the baseline, we selected the Tree-Ring [20] and Stable Signature [24] watermarking algorithms for several reasons. In this paper, we aim to present a robust watermarking method capable of encrypting multiple messages. Tree-Ring [20] is considered a state-of-the-art algorithm when it comes to robustness to various attacks [21]. To our knowledge, Stable Signature is the only method capable of encrypting messages non-post-generation, where watermarking happens during the generation process [24]. Finally, **METR** and **METR++** build upon these existing watermarking algorithms.

We compare the models based on three main criteria: the accuracy of watermark detection, the accuracy of watermarked message decryption, and image quality, since we aim to ensure that the encrypted message does not degrade the generated image. The accuracy of watermark detection is measured by False Positive Rate (FPR), True Positive Rate (TPR), Area Under Curve (AUC) of Receiver Operator Characteristic (ROC), TPR value when FPR equals to 1%, denoted as "TPR@1%FPR". The quality of message detection is assessed using Bit Accuracy, Word Accuracy (the accuracy of fully detected messages, as proposed in [21]), and the detection resolution metric described in Section 3.2. To assess image quality, we use **FID** (Fréchet Inception Distance) [36] and **CLIP** score [37]. FID is calculated by comparing the generated watermarked image to its corresponding ground-truth pair from the dataset, denoted as **FID gt**, or to a generated image created with the same prompt but without a watermark, denoted as **FID gen**. Comparison with the ground truth image indicates overall quality, while comparison with the generated image illustrates the impact of watermark encryption on the generation process. For the CLIP score, we measure the cosine similarity between the embedding of the watermarked image and the embedding of the reference prompt, following OpenCLIP-ViT/G [38].

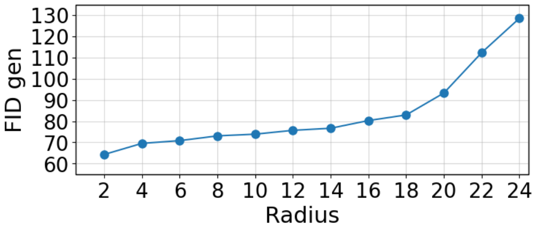


Figure 7: **METR** evaluation on image quality with different radii

We utilized the MSCOCO-5000 dataset [39] to evaluate **FID**. This dataset consists of 5000 paired images and prompts. To assess watermark detection accuracy, message detection quality, and image quality using **CLIP**, the images were generated using a subset of 1000 randomly selected prompts from the Stable Diffusion prompts [40].

4.2. METR Evaluation

In this subsection, we describe a set of experiments that compare **METR** with *Tree-Ring*. We focus on evaluating their robustness to white-box attacks, including ones that utilize generative models, as described in Section 2.4, watermark detection accuracy and the overall quality of the generated images.

To select proper **METR** parameters, we first searched for the optimal message scale S with the algorithm described in Section 3.2 in range $60 \leq S \leq 160$ with multiple different radii. The resulting optimal value was always between 80 and 100. For the rest of our experiments, S was set to 100 unless specified otherwise. Regarding message radius r, we evaluated image quality for different radii on a subset of the MSCOCO-5000 dataset with 500 images. See Figure 7 for results.

It is clear that keeping the radius as small as possible benefits the quality of the resulting image. However, we argue that increasing the radius to 16 can still maintain acceptable overall quality. The benefit of doing so is that, this way, **METR** can encode up to $2^{16} = 65536$ messages. In our experiments, we set $r = 10$, which allowed us to encode 1024 messages with almost no loss in quality.

Table 1
Detection metrics for *Tree-Ring* and **METR**. R_{det} denotes detection resolution metric. Note that for most of the white-box attacks **METR** is as resilient as, or even more resilient than, *Tree-Ring*.

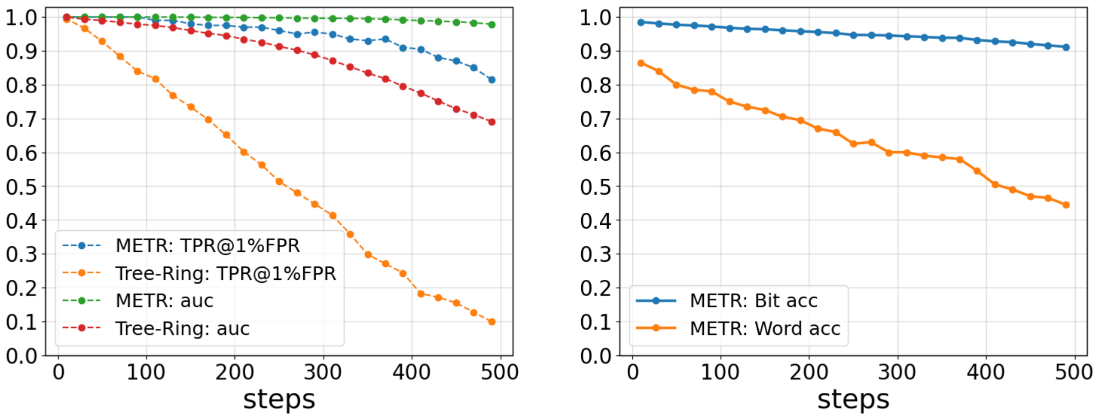
Attack	*Tree-Ring*	**METR**	**METR**		
	AUC (↑)	**AUC** (↑)	**Bit Acc** (↑)	**Word Acc** (↑)	$\mathbf{R}_{\text{det}}$ (↑)
No attack	1.000 (+0.0%)	1.000 (+0.0%)	0.991	0.910	42.7
VAE [34] $q = 1$	0.999 (-0.1%)	0.999 (-0.1%)	0.968	0.747	22.4
diff, 150 steps	0.952 (-4.8%)	**0.999(-0.1%)**	0.967 (-3.3%)	0.732	21.3
rotate 75	**0.959 (-4.1%)**	0.907 (-9.3%)	0.694	0.033	5.1
brightness 6.0	0.994 (-0.6%)	**0.995 (-0.5%)**	0.912	0.463	26.0
noise $\sigma = 0.1$	**0.941 (-5.9%)**	0.833 (-16.7%)	0.695	0.134	7.9
blur $r = 4$	1.000 (-0.0%)	1.000 (-0.0%)	0.979	0.805	34.1
crop 0.75	**0.911 (-8.9%)**	0.807 (-19.3%)	0.668	0.020	5.4
JPEG 25	**0.995 (-0.5%)**	0.992 (-0.8%)	0.953	0.719	27.0

Detection accuracy under image transformation attacks. The detection metrics for cases with white-box attacks [21] involving **METR** and *Tree-Ring* are presented in Table 1. "B acc" and "W acc" represent Bit Accuracy and Word Accuracy, respectively. Since *Tree-Ring* cannot encode messages, these metrics are only calculated for **METR**. As demonstrated by the results in Table 1, **METR** is as resilient as, or even more resilient than, *Tree-Ring* against most image transformations. Both algorithms find rotation and cropping to be the most challenging types of attacks.

Table 2
Message decryption accuracy for **METR**++ and its components: **METR** and *Stable Signature* watermarks. Note that the word accuracy of **METR**++ is constrained by the lower word accuracy of either the **METR** or *Stable Signature* watermarks.

Attack	METR: ft VAE		Stable-Signature		METR++	
	Bit Acc	Word Acc	Bit Acc	Word Acc	Bit Acc	Word Acc
None	0.991	0.914	0.995	0.843	0.995	0.772
VAE $q = 1$	0.970	0.760	0.490	0.000	0.572	0.000
diff, 150 steps	0.966	0.719	0.477	0.000	0.561	0.000
rot 75	0.687	0.029	0.547	0.000	0.572	0.000
bright 6.0	0.909	0.460	0.904	0.264	0.905	0.209
noise $\sigma = 0.1$	0.694	0.137	0.539	0.000	0.565	0.000
blur $r = 4$	0.981	0.816	0.419	0.000	0.516	0.000
crop 0.75	0.669	0.023	0.982	0.569	0.928	0.013
JPEG 25	0.941	0.672	0.769	0.000	0.799	0.000

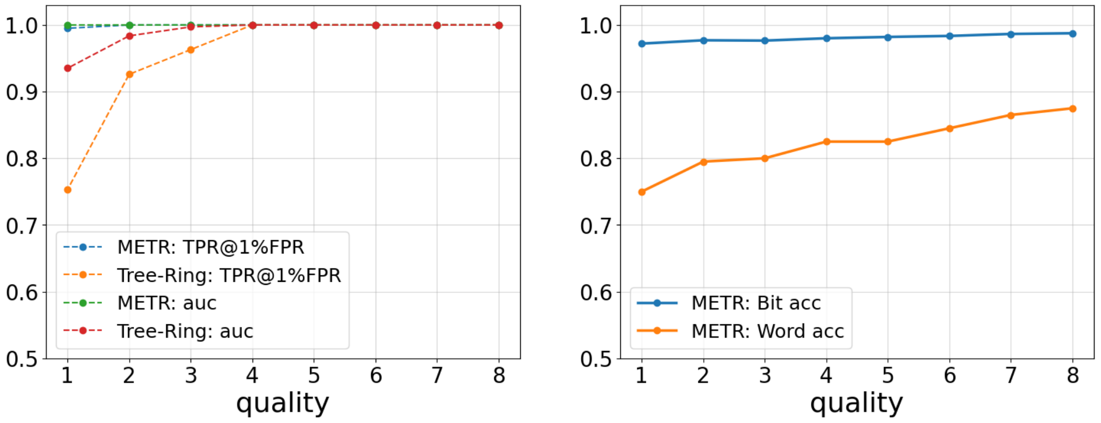
Detection accuracy under generative models' attacks. We performed a diffusion attack using the base version of Stable-Diffusion 2.1 [11] and a VAE attack using the model [34] with a quality hyperparameter $q = 1$. The detection metrics for images generated with *Tree-Ring* and **METR** watermark, both with the diffusion attack, can be seen in Figure 8. As one can see in the chart, the AUC and TPR@1%FPR for METR are higher than those for *Tree-Ring*, and decrease with the number of diffusion steps more slowly. Similar results are shown for the VAE attack in Figure 9, parametrized by q. One can see that *Tree-Ring* detection metrics are below 1 until $q = 4$, while **METR** is robust to this attack and the watermark is almost always detected correctly. The *Tree-Ring* algorithm is not capable of message encryption, and thus word and bit accuracy are shown only for the **METR** algorithm.



(a) Watermark detection

(b) Watermark decryption

Figure 8: Watermark detection for *Tree-Ring* and **METR** with the Diffusion attack [34]. Note that the watermark detection metrics for **METR** are always higher than for *Tree-Ring* and decrease with the number of diffusion steps more slowly.



(a) Watermark detection

(b) Watermark decryption

Figure 9: Robustness of *Tree-Ring* and **METR** with a VAE attack [34]. Note that the detection metrics for *Tree-Ring* are below 1 until $q = 4$, while for **METR** they always equal 1.

Image quality. The results of comparing **METR** with *Tree-Ring* are presented in Table 3. For the experiment, we sampled a random message for the **METR** algorithm. It can be seen that the quality of images generated with the **METR** watermark is close to those with the *Tree-Ring* watermark. In terms of the CLIP score, the performance of *Tree-Ring* is similar to that of an image without a watermark, while it drops by 2% for **METR**. When it comes to the FID score, the relative increase for images generated without a watermark and those with *Tree-Ring* and **METR** watermarks are 1.7% and 3.2%, respectively.

In conclusion, the **METR** algorithm demonstrates high resilience to all white-box attacks. The accuracy of watermark detection, even without any attacks or under white-box attack conditions, is very close to the detection accuracy achieved by *Tree-Ring*, and surpasses *Tree-Ring* for attacks with generative models [21]. As for image quality, we noticed only a slight decrease with the **METR**

Table 3
The image quality for images without watermarks is assessed using *Tree-Ring* and **METR**. Note that for watermark-free images, those generated using *Tree-Ring* produce results that are nearly identical to those generated with **METR**, although *Tree-Ring* scores are slightly higher.

Method	Message	**FID gen** (↓)	**FID gt** (↓)	**CLIP** (↑)
No WM	-	-	25.570 (+0.0%)	0.364 (-0.0%)
Tree-Ring	-	10.283 (+0.0%)	26.007 (+1.7%)	0.364 (-0.0%)
METR	+	10.971 (+6.7%)	26.398 (+3.2%)	0.357 (-1.9%)
METR++	+	11.017 (+6.9%)	25.206 (-1.0%)	0.365 (+0.3%)

algorithm when compared to *Tree-Ring*, with less than a 2% difference. However, **METR** makes it possible to encode multiple unique messages in a watermark. The accuracy of the decrypted messages showed high resilience to the majority of white-box attacks, with exceptions being rotation and cropping. The word accuracy for messages without any attack reached 0.91.

4.3. METR++ Evaluation

In this section, we perform a series of experiments to evaluate the robustness of **METR++** against white-box attacks. We also compare its detection metrics with those of **METR** and the *Stable Signature* watermark [24].

As previously detailed in Section 3.3, METR++ consists of the METR watermark and a fine-tuned VAE decoder designed to encrypt a 48-bit *Stable Signature* [24] message. Therefore, our evaluation begins by investigating whether the inclusion of the METR watermark in the Fourier space of the latent noise decreases detection resilience. To assess this, we fine-tune the VAE decoder using images drawn from standard latent noise and a distribution in which the METR watermark is embedded into the latent noise space. We decode the messages encoded into the images using both the standard VAE decoder and the one that was fine-tuned on images with the METR watermark. The results are presented in Table 4. The term $\mathcal{N}$-METR refers to the VAE decoder that was fine-tuned on images sampled from a normal distribution and subsequently used to work with images with the METR watermark. It can be observed that the bit accuracy of the decrypted *Stable Signature* message in **METR++** is not affected by whether the VAE decoder has been fine-tuned on images sampled from latent noise with the embedded METR watermark. When it comes to the attacks, the bit accuracy of the decrypted *Stable Signature* message remains almost constant, showing a change of less than 1% compared to basic *Stable Signature*.

Table 4
The bit accuracy of the Stable-Signature watermark. Note that the bit accuracy of the decrypted *Stable Signature* message from the images containing a **METR** watermark remains nearly constant, with a variation of less than 1% compared to the images without a **METR** watermark.

VAE decoder	no attack	crop 0.1	bright 2	jpeg 50
$\mathcal{N}$-$\mathcal{N}$	0.997	0.969 (+0.0%)	0.990 (+0.0%)	0.870 (+0.0%)
$\mathcal{N}$-METR	0.997	0.972 (+0.3%)	0.991 (+0.1%)	0.867 (-0.3%)
METR-$\mathcal{N}$	0.997	0.971 (+0.3%)	0.989 (-0.1%)	0.866 (-0.4%)
METR-METR	0.997	0.974 (+0.5%)	0.990 (-0.0%)	0.863 (-0.8%)

The detection accuracy of **METR++** consists of the detection accuracies for both the *Stable Signature* and **METR** watermarks. Table 2 presents the results of the detection accuracy evaluation. The term "METR: ft VAE" refers to the accuracy of detecting the METR watermark when processed through the **METR++** pipeline. "Stable-Signature" refers to the accuracy of decrypting the referenced watermark using the **METR++** pipeline. Lastly, "**METR++** " denotes the overall detection accuracy of the complete

METR++ message, which includes both the **METR** and *Stable Signature* messages. The detection accuracy of the **METR** watermark remains unaffected by the **METR++** pipeline, as the image decoded by the fine-tuned decoder closely resembles the one generated by the original VAE decoder. Consequently, both the Bit and Word accuracy for **METR** watermark detection, as previously shown in Table 1, demonstrate robustness against most white-box attacks, aside from rotation and cropping. On the other hand, *Stable Signature* is vulnerable to white-box attacks, which in turn impacts the Word Accuracy of **METR++**. In Table 1, we highlighted a higher Word Accuracy between the **METR** and *Stable Signature* watermarks. It is important to note that the robustness of **METR++**, in terms of word accuracy, is limited to the lesser robustness of the two watermarks since it requires the correct decryption of both and in terms of bit accuracy it is highly correlated with *Stable Signature* part, because it has more bits, than **METR** part of **METR++**.

In terms of the potential number of messages, we use 58 bits for each message therefore, it is possible to encode 2^{58} unique messages. When considering the practical application of either **METR** or **METR++**, it is important to evaluate the trade-off between the capability to encode numerous unique messages and the resilience of the watermarking method against white-box attacks.

5. Future Work

In this study, we assessed the detection accuracy of **METR** and **METR++** against several white-box attacks. **METR** demonstrated high resilience to most of the attacks, including state-of-the-art ones. We also propose researching methods to fine-tune the weights of the generative model to enhance the robustness of the **METR** watermarking algorithm. When it comes to **METR++**, this method showed low robustness to most attacks due to its word accuracy robustness being constrained by the lowest detection accuracy among the two messages. The *Stable Signature* watermark message is generally less robust to attacks than the **METR** watermark message. Therefore, to increase the encoding capacity of the **METR** watermarking algorithm, we suggest exploring alternative watermarking methods that could replace *Stable Signature*, or investigating *Stable Signature* modifications within **METR++** that improve its detection accuracy under attacks.

Another area of research that needs further exploration is the impact of specific messages on the quality of the generated images. Our findings indicate that the quality of images generated using an average **METR** message is comparable to those produced with the *Tree-Ring* watermark. However, in real-world applications, it is essential to examine various combinations of binary values to ascertain whether certain messages with "extreme values" might significantly affect image quality. This consideration could be crucial for commercial projects aiming to ensure consistent image generation quality for all their users.

6. Conclusion

In this paper, we introduced the **METR** watermarking algorithm, which can be applied to any diffusion model architecture with a non-probabilistic sampler (*e.g.,* DDIM [30]). Building upon the *Tree-Ring* watermarking algorithm, **METR** retains its robustness in detection under white-box attacks and high image quality. The most significant advantage of **METR** is its ability to encrypt multiple unique messages without the need for model's weights fine-tuning. This positions the proposed watermarking algorithm as one that can be considered the current leading watermarking algorithm in terms of robustness, image quality, and message encoding capacity. We also propose an extension of **METR**, named **METR++**, which is specifically tailored to Latent Diffusion Models and requires additional fine-tuning of a VAE decoder for each new user group. **METR++** increases the potential number of encoded messages by the factor of fine-tuned VAE decoders. Comparing **METR** with its extension **METR++** shows that the decision of which algorithm to use for practical applications should balance the total number of messages that can be encoded and the watermark's overall robustness to attacks. Implementation of our work can be found at https://github.com/deepvk/metr.

References

[1] CivitAI, Civitai, 2024. URL: https://civitai.com/.

[2] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, M. Chen, Hierarchical text-conditional image generation with clip latents, 2022. arXiv:2204.06125.

[3] F. Khader, G. Mueller-Franzes, S. T. Arasteh, T. Han, C. Haarburger, M. Schulze-Hagen, P. Schad, S. Engelhardt, B. Baessler, S. Foersch, J. Stegmaier, C. Kuhl, S. Nebelung, J. N. Kather, D. Truhn, Medical diffusion: Denoising diffusion probabilistic models for 3d medical image generation, 2023. arXiv:2211.03364.

[4] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, T. Goldstein, Diffusion art or digital forgery? investigating data replication in diffusion models, 2022. arXiv:2212.03860.

[5] S. Lee, G. Gu, S. Park, S. Choi, J. Choo, High-resolution virtual try-on with misalignment and occlusion-handled conditions, 2022. arXiv:2206.14180.

[6] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444. URL: https://doi.org/10.1038/nature14539. doi:10.1038/nature14539.

[7] D. P. Kingma, M. Welling, Auto-encoding variational bayes, 2022. arXiv:1312.6114.

[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, 2014. arXiv:1406.2661.

[9] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, 2020. arXiv:2006.11239.

[10] A. Razzhigaev, A. Shakhmatov, A. Maltseva, V. Arkhipkin, I. Pavlov, I. Ryabov, A. Kuts, A. Panchenko, A. Kuznetsov, D. Dimitrov, Kandinsky: an improved text-to-image synthesis with image prior and latent diffusion, 2023. arXiv:2310.03502.

[11] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, 2022. arXiv:2112.10752.

[12] Pinterest, Pinterest, 2024. URL: https://www.pinterest.com/.

[13] C. Novelli, F. Casolari, P. Hacker, G. Spedicato, L. Floridi, Generative ai in eu law: Liability, privacy, intellectual property, and cybersecurity, 2024. arXiv:2401.07348.

[14] A. Bashardoust, S. Feuerriegel, Y. R. Shrestha, Comparing the willingness to share for human-generated vs. ai-generated fake news, 2024. arXiv:2402.07395.

[15] D. Benalcazar, J. E. Tapia, S. Gonzalez, C. Busch, Synthetic id card image generation for improving presentation attack detection, 2022. arXiv:2211.00098.

[16] N. Raman, S. Shah, M. Veloso, Synthetic document generator for annotation-free layout recognition, Pattern Recognition 128 (2022) 108660. URL: http://dx.doi.org/10.1016/j.patcog.2022.108660. doi:10.1016/j.patcog.2022.108660.

[17] Y. Qu, X. Shen, X. He, M. Backes, S. Zannettou, Y. Zhang, Unsafe diffusion: On the generation of unsafe images and hateful memes from text-to-image models, 2023. arXiv:2305.13873.

[18] I. Cox, M. Miller, J. Bloom, J. Fridrich, T. Kalker, Digital Watermarking and Steganography, 2nd ed., Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.

[19] C.-C. Chang, P. Tsai, C.-C. Lin, Svd-based digital image watermarking scheme, Pattern Recognition Letters 26 (2005) 1577–1586. URL: https://www.sciencedirect.com/science/article/pii/S0167865505000140. doi:10.1016/j.patrec.2005.01.004.

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

[21] X. Zhao, K. Zhang, Z. Su, S. Vasan, I. Grishchenko, C. Kruegel, G. Vigna, Y.-X. Wang, L. Li, Invisible image watermarks are provably removable using generative ai, 2023. arXiv:2306.01953.

[22] Y. Li, H. Wang, M. Barni, A survey of deep neural network watermarking techniques, 2021. arXiv:2103.09274.

[23] R. B. Wolfgang, E. J. Delp, A watermark for digital images, in: Proceedings of 3rd IEEE International Conference on Image Processing, volume 3, IEEE, 1996, pp. 219–222.

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

[25] J. Zhu, R. Kaplan, J. Johnson, L. Fei-Fei, Hidden: Hiding data with deep networks, 2018.

arXiv:1807.09937.

[26] K. A. Zhang, L. Xu, A. Cuesta-Infante, K. Veeramachaneni, Robust invisible video watermarking with attention, 2019. arXiv:1909.01285.

[27] M. Tancik, B. Mildenhall, R. Ng, Stegastamp: Invisible hyperlinks in physical photographs, 2020. arXiv:1904.05343.

[28] Z. Jiang, J. Zhang, N. Z. Gong, Evading watermark based detection of ai-generated content, 2023. arXiv:2305.03807.

[29] Y. Liu, Z. Li, M. Backes, Y. Shen, Y. Zhang, Watermarking diffusion model, 2023. arXiv:2305.12502.

[30] J. Song, C. Meng, S. Ermon, Denoising diffusion implicit models, 2022. arXiv:2010.02502.

[31] P. B. Patnaik, The non-central χ^2- and F-distribution and their applications, Biometrika 36 (1949) 202–232. URL: https://www.jstor.org/stable/2332542. doi:10.2307/2332542.

[32] P. Glasserman, Monte Carlo Methods in Financial Engineering, volume 53 of *Stochastic Modelling and Applied Probability*, Springer, New York, NY, 2003. URL: http://link.springer.com/10.1007/978-0-387-21617-1. doi:10.1007/978-0-387-21617-1.

[33] S. Peng, Y. Chen, C. Wang, X. Jia, Intellectual property protection of diffusion models via the watermark diffusion process, 2023. arXiv:2306.03436.

[34] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, N. Johnston, Variational image compression with a scale hyperprior, 2018. arXiv:1802.01436.

[35] E. O. Brigham, R. E. Morrow, The fast fourier transform, IEEE Spectrum 4 (1967) 63–70. doi:10.1109/MSPEC.1967.5217220.

[36] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. arXiv:1706.08500.

[37] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, 2021. arXiv:2103.00020.

[38] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, J. Jitsev, Reproducible scaling laws for contrastive language-image learning, 2022. arXiv:2212.07143.

[39] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.

[40] Gustavosta, Stable diffusion prompts, 2022. URL: https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts.