# The Stable Signature: Rooting Watermarks in Latent Diffusion Models

Pierre Fernandez[1,2] Guillaume Couairon[1,3] Hervé Jégou[1] Matthijs Douze[1] Teddy Furon[2*]

[1]Meta AI [2]Centre Inria de l'Université de Rennes [3]Sorbonne University

## Abstract

*Generative image modeling enables a wide range of applications but raises ethical concerns about responsible deployment. We introduce an active content tracing method combining image watermarking and Latent Diffusion Models. The goal is for all generated images to conceal an invisible watermark allowing for future detection and/or identification. The method quickly fine-tunes the latent decoder of the image generator, conditioned on a binary signature. A pre-trained watermark extractor recovers the hidden signature from any generated image and a statistical test then determines whether it comes from the generative model. We evaluate the invisibility and robustness of the watermarks on a variety of generation tasks, showing that the Stable Signature is robust to image modifications. For instance, it detects the origin of an image generated from a text prompt, then cropped to keep 10% of the content, with 90+% accuracy at a false positive rate below $10^{-6}$.*

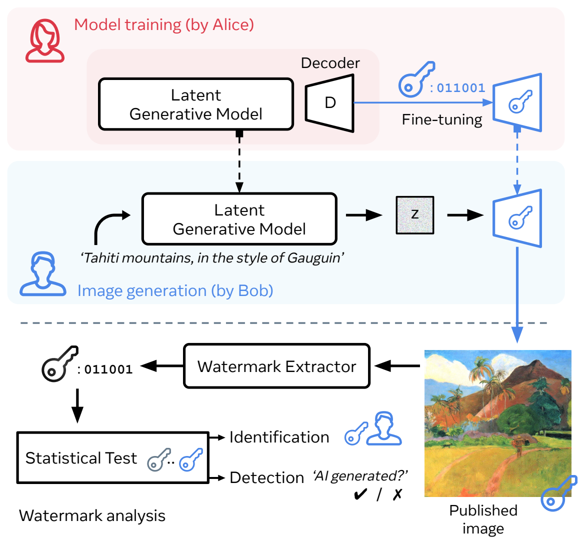



Figure 1. Overview. The latent decoder can be fine-tuned to preemptively embed a signature into all generated images.

## 1. Introduction

Recent progress in generative modeling and natural language processing enable easy creation and manipulation of photo-realistic images, such as with DALL·E 2 [64] or Stable Diffusion [68]. They have given birth to many image edition tools like ControlNet [104], Instruct-Pix2Pix [7], and others [13, 28, 71], that are becoming mainstream creative tools for artists, designers, and the general public.

While this is a great step forward for generative AI, it also undermines confidence in the authenticity or veracity of photo-realistic images. Indeed, methods for photo-realistic image edition existed before, but generative AI significantly lowers the barriers to convincing synthetic image generation and edition (*e.g.* a generated picture recently won an art competition [29]). This raises new risks like deep fakes, impersonation or copyright usurpation [8, 17]. A tool to determine that images are AI-generated would make it easier to ensure their compliance with ethical standards and to remove them from certain platforms.

*Work supported by ANR / AID under Chaire SAIDA ANR-20-CHIA-0011.
Correspondance to pfz@meta.com

A baseline solution to identify generated images is forensics, *i.e.* methods to detect generated/manipulated images passively (the image is not modified for identification). An active baseline is to apply existing watermarking methods after the image generation. Watermarking invisibly embeds a secret message into the image, which can then be extracted and used to identify the image. This has several drawbacks. If the model leaks or is open-sourced, the post-generation watermarking can be removed trivially. The open source Stable Diffusion [70] is a case in point, since removing the watermark amounts to commenting out a single line in the source code.

Our Stable Signature method merges watermarking into the generation process itself, without any architectural change. It adjusts the pre-trained generative model such that all the images it produces conceal a given watermark. There are several advantages to this approach [46, 99]. It does not require additional processing of the generated image, which makes the watermarking computationally lighter, straightforward, and secure. Model providers could deploy their models to different user groups with a unique watermark, and monitor that they are used in a responsible manner.

They could give art platforms, news outlets and other sharing platforms the ability to detect when an image has been generated by their AI.

We focus on Latent Diffusion Models (LDM) [68] that can perform a wide range of generative tasks. We show that simply fine-tuning a small part of the generative model – the decoder that generates images from the latent vectors – is enough to natively embed a watermark into generated images. Stable Signature does not require an architectural change and does not modify the diffusion process. Hence it is to our knowledge compatible with all LDM-based generative methods [7, 13, 62, 71, 104]. The fine-tuning stage is performed by back-propagating a combination of a perceptual image loss and a message decoding loss from a watermark extractor back to the LDM decoder. We pre-train the extractor with a simplified version of the deep watermarking method HiDDeN [108].

We create an evaluation benchmark close to real world situations where images may be edited. The tasks are: detection of AI generated images, tracing models from their generations. For instance, we detect 90% of images generated with the generative model, even if they are cropped to 10% of their original size, while flagging only one false positive every $10^6$ images. To ensure that the model's utility is not weakened, we show that the FID [34] score of the generation is not affected and that the generated images are perceptually indistinguishable from the ones produced by the original model. This is done over several tasks involving LDM (text-to-image, inpainting, edition, etc.).

As a summary, (1) we efficiently merge watermarking into the generation process of LDMs, in a way that is compatible with most of the LDM-based generative methods; (2) we demonstrate how it can be used to detect and trace generated images, through a real-world evaluation benchmark; (3) we compare to post-hoc watermarking methods, showing that it is competitive while being more secure and efficient, and (4) evaluate robustness to intentional attacks.

## 2. Related Work

**Image generation** has long been dominated by GANs, still state-of-the-art on many datasets [37, 38, 39, 76, 87]. Transformers have also been successfully used for modeling image [66, 19] or video [79] distributions, providing higher diversity at the expense of increased inference time. Images are typically converted to token lists using vector-quantized architectures [20, 67, 96], relying on an image decoder.

Diffusion models [18, 36, 58, 80] are a huge improvement in text-conditional image generation, current models synthesize high-resolution photo-realistic images for a wide variety of text prompts [3, 35, 65, 69, 74]. They can also perform conditional image generation – like inpainting or text-guided image editing – by fine-tuning the diffusion model with additional conditioning, *e.g*. masked input image, segmentation map, etc. [50, 73]. Because of their iterative denoising algorithm, diffusion models can also be adapted for image editing in a zero-shot fashion by guiding the generative process [13, 33, 40, 56, 85, 92]. All these methods, when combined with Stable Diffusion, operate in the latent rather than image space, requiring a latent decoder to produce an RGB image.

**Detection of AI-generated/manipulated images** is notably active in the context of deep-fake detection [32, 107]. Many works focus on the detection of GAN-generated images [10, 31, 89, 106]. One approach is to detect inconsistencies, via lights, perspective or physical objects [21, 22, 45, 52, 88]. These approaches are restricted to photo-realistic images or faces, artworks not intended to be physically correct are not covered.

Other approaches track traces left by the generators in the spatial [54, 97] or frequency [26, 106] domains. There are extensions to diffusion models in recent works [12, 77] that show encouraging results. However purely relying on forensics and passive detection is limiting, *e.g*. the best performing method to our knowledge [12] is able to detect 50% of generated images for a false positive rate around 1/100: if a user-generated content platform were to receive 1 billion images every day, it would wrongly flag 10 million images to detect only half of the generated images. Besides, passive techniques cannot trace images from different versions of the same model, in contrast with watermarking.

**Image watermarking** has long been studied in the context of tracing and intellectual property protection [14]. More recently, deep learning encoder/extractors like HiDDeN [2, 44, 51, 100, 108] or iterative methods by Vukotić *et al*. [25, 42, 86] showed competitive results in terms of robustness to a wide range of transformations, namely geometric ones.

In the specific case of **generative models**, some works aim to watermark the training set on which the generative model is learned [98]. It is highly inefficient since every new message to embed requires a new training pipeline. Merging the watermarking and the generative process is a recent idea [23, 46, 59, 63, 93, 99, 102], that is closer to the model watermarking literature [84]. They suffer from two strong limitations. First, these methods only apply to GANs, while LDM are progressively replacing them for most applications. Second, watermarking is incorporated in the training process of the GAN from the start. This strategy is unsustainable because the generative model training is more and more costly[1]. Our work shows that a quick fine-tuning of the latent decoder part of the generative model is enough to achieve a good watermarking performance, provided that the watermark extractor is well chosen.

[1] Stable Diffusion training costs ∼\$600k of cloud compute (Wikipedia).

## 3. Problem Statement & Background

Figure 1 shows a model provider *Alice* who deploys a latent diffusion model to users *Bobs*. Stable Signature embeds a binary signature into the generated images. This section derives how Alice can use this signature for two scenarios:

- *Detection: "Is it generated by my model?"*. Alice detects if an image was generated by her model. Generated images should be flagged as reliably as possible, while controlling the probability of flagging a natural image.
- *Identification: "Who generated this image?"*. Alice monitors who created each image, while avoiding to mistakenly identifying a Bob.

### 3.1. Image watermarking for detection

Alice embeds a $k$-bit binary signature into the generated images. The watermark extractor then decodes messages from the images it receives and detects when the message is close to Alice's signature. An example application is to block AI-generated images on a content sharing platform.

**Statistical test** Let $m \in \{0,1\}^k$ be Alice's signature. We extract the message $m'$ from an image $x$ and compare it to $m$. As done in previous works [46, 98], the detection test relies on the number of matching bits $M(m, m')$: if

$$M\left(m, m'\right) \geq \tau \quad \text{where} \quad \tau \in \{0, \ldots, k\}, \tag{1}$$

then the image is flagged. This provides a level of robustness to imperfections of the watermarking.

Formally, we test the statistical hypothesis $H_1$: "$x$ was generated by Alice's model" against the null hypothesis $H_0$: "$x$ was not generated by Alice's model". Under $H_0$ (*i.e.* for vanilla images), we assume that bits $m'_1, \ldots, m'_k$ are (i.i.d.) Bernoulli random variables with parameter $0.5$. Then $M(m, m')$ follows a binomial distribution with parameters $(k, 0.5)$. We verify this assumption experimentally in App. B.5. The False Positive Rate (FPR) is the probability that $M(m, m')$ takes a value bigger than the threshold $\tau$. It is obtained from the CDF of the binomial distribution, and a closed-form can be written with the regularized incomplete beta function $I_x(a; b)$:

$$\text{FPR}(\tau) = \mathbb{P}\left(M > \tau | H_0\right) = I_{1/2}(\tau + 1, k - \tau). \tag{2}$$

### 3.2. Image watermarking for identification

Alice now embeds a signature $m^{(i)}$ drawn randomly from $\{0,1\}^k$ into the model distributed to Bob$^{(i)}$ (for $i = 1 \cdots N$, with $N$ the number of Bobs). Alice can trace any misuse of her model: generated images violating her policy (gore content, deepfakes) are linked back to the specific Bob by comparing the extracted message to Bobs' signatures.

**Statistical test** We compare the message $m'$ from the watermark extractor to $\left(m^{(1)}, \ldots, m^{(N)}\right)$. There are now $N$ detection hypotheses to test. If the $N$ hypotheses are rejected, we conclude that the image was not generated by any of the models. Otherwise, we attribute the image to $\text{argmax}_{i=1..N} M\left(m', m^{(i)}\right)$. With regards to the detection task, false positives are more likely since there are $N$ tests. The global FPR at a given threshold $\tau$ is:

$$\text{FPR}(\tau, N) = 1 - (1 - \text{FPR}(\tau))^N \approx N.\text{FPR}(\tau). \tag{3}$$

Equation (3) (resp. (2)), is used reversely: we find threshold $\tau$ to achieve a required FPR for identification (resp. detection). Note that these formulae hold only under the assumption of i.i.d. Bernoulli bits extracted from vanilla images. This condition is enforced in the next section.

## 4. Method

Stable Signature modifies the generative network so that the generated images have a given signature through a fixed watermark extractor. It is trained in two phases. First, we create the watermark extractor network $\mathcal{W}$. We then fine-tune the LDM decoder $\mathcal{D}$, such that all generated images yield a given signature through $\mathcal{W}$.

### 4.1. Pre-training the watermark extractor

We use HiDDeN [108], a classical method in the deep watermarking literature. It jointly optimizes the parameters of watermark encoder $\mathcal{W}_E$ and extractor network $\mathcal{W}$ to embed $k$-bit messages into images, robustly to transformations that are applied during training. We discard $\mathcal{W}_E$ after training, since only $\mathcal{W}$ serves our purpose.

Formally, $\mathcal{W}_E$ takes as inputs a cover image $x_o \in \mathbb{R}^{W \times H \times 3}$ and a $k$-bit message $m \in \{0,1\}^k$. Similar to

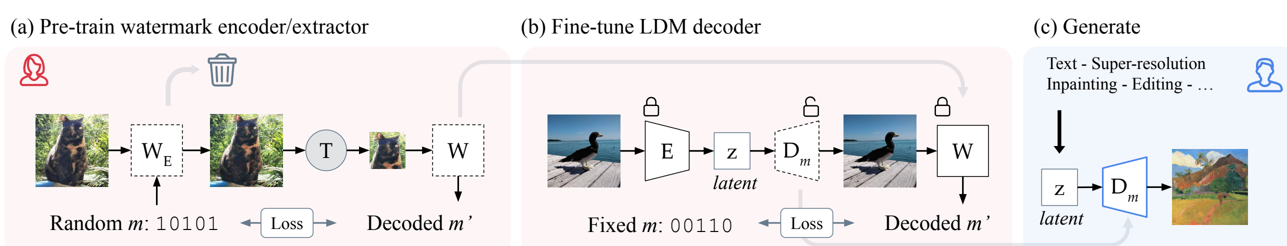



Figure 2. Steps of the method. (a) We pre-train a watermark encoder $\mathcal{W}_E$ and extractor $\mathcal{W}$, to extract binary messages. (b) We fine-tune the decoder $\mathcal{D}$ of the LDM's auto-encoder with a fixed signature $m$ such that all the generated images (c) lead to $m$ through $\mathcal{W}$.

ReDMark [2], $\mathcal{W}_E$ outputs a residual image $\delta$ of the same size as $x_o$, that is multiplied by a factor $\alpha$ to produce watermarked image $x_w = x_o + \alpha\delta$. At each optimization step an image transformation $T$ is sampled from a set $\mathcal{T}$ that includes common image processing operations such as cropping and JPEG compression[2]. A "soft" message is extracted from the transformed image: $m' = \mathcal{W}(T(x_w))$ (at inference time, the decoded message is given by the signs of the components of $m'$). The *message loss* is the Binary Cross Entropy (BCE) between $m$ and the sigmoid $\sigma(m')$:

$$\mathcal{L}_m = -\sum_{i=1}^{k} m_i \cdot \log \sigma(m'_i) + (1 - m_i) \cdot \log(1 - \sigma(m'_i)).$$

The network architectures are kept simple to ease the LDM fine-tuning in the second phase. They are the same as HiDDeN [108] (see App. A.1) with two changes.

First, since $\mathcal{W}_E$ is discarded, its perceptual quality is not as important, so the perceptual loss or the adversarial network are not needed. Instead, the distortion is constrained by a $\tanh$ function on output of $\mathcal{W}_E$ and by the scaling factor $\alpha$. This improves the bit accuracy of the recovered message and makes it possible to increase its size $k$.

Second, we observed that $\mathcal{W}$'s output bits for vanilla images are correlated and highly biased, which violates the assumptions of Sec. 3.1. Therefore, to get closer to i.i.d. dimensions we remove the bias and decorrelate the outputs of $\mathcal{W}$ by applying a PCA whitening transformation (more details in App. A.1).

### 4.2. Fine-tuning the generative model

In LDM, the diffusion happens in the latent space of an auto-encoder. The latent vector $z$ obtained at the end of the diffusion is input to decoder $\mathcal{D}$ to produce an image. Here we fine-tune $\mathcal{D}$ such that the image contains a given message $m$ that can be extracted by $\mathcal{W}$. Stable Signature is compatible with many generative tasks, since modifying only $\mathcal{D}$ does not affect the diffusion process.

First, we fix the signature $m = (m_1, \ldots, m_k) \in \{0, 1\}^k$. The fine-tuning of $\mathcal{D}$ into $\mathcal{D}_m$ is inspired by the original training of the auto-encoder in LDM [68].

Training image $x \in \mathbb{R}^{H\times W\times 3}$ is fed to the LDM encoder $\mathcal{E}$ that outputs activation map $z = \mathcal{E}(x) \in \mathbb{R}^{h\times w\times c}$, downsampled by a power-of-two factor $f = H/h = W/w$. The decoder reconstructs an image $x' = \mathcal{D}_m(z)$ and the extractor recovers $m' = \mathcal{W}(x')$. The *message loss* is the BCE between $m'$ and the original $m$: $\mathcal{L}_m = \mathrm{BCE}(\sigma(m'), m)$.

In addition, the original decoder $\mathcal{D}$ reconstructs the image without watermark: $x'_o = \mathcal{D}(z)$. The *image perceptual loss* $\mathcal{L}_\mathrm{i}$ between $x'$ and $x'_o$, controls the distortion. We use the Watson-VGG perceptual loss introduced by Czolbe *et al*. [15], an improved version of LPIPS [105]. It is essential that the decoder learns luminance and contrast masking to add less visible watermarks. The weights of $\mathcal{D}_m$ are optimized to minimize

$$\mathcal{L} = \mathcal{L}_\mathrm{m} + \lambda_\mathrm{i}\, \mathcal{L}_\mathrm{i}. \tag{4}$$

This is done over 100 iterations with the AdamW optimizer [49] and batch of size 4, *i.e.* the fine-tuning sees *fewer than 500 images* and takes *one minute on a single GPU*. The learning rate follows a cosine annealing schedule with 20 iterations of linear warmup to $10^{-4}$ and decays to $10^{-6}$. The factor $\lambda_\mathrm{i}$ in (4) is set to 0.2 by default.

## 5. Text-to-Image Watermarking Performance

This section discusses our method's detection and identification capability for images generated by a Stable-Diffusion-like model [68][3]. We apply generative models watermarked with 48-bit signatures on prompts of the MS-COCO [47] validation set. We evaluate detection and identification on the outputs, as illustrated in Figure 1.

We evaluate their robustness to different transformations applied to generated images: strong cropping (10% of the image remaining), brightness shift (strength factor 2.0), as well as a combination of crop 50%, brightness shift 1.5 and JPEG 80. This covers typical geometric and photometric edits (see Fig. 5 for visual examples).

The detection rates are partly obtained from experiments and partly by extrapolating small-scale measurements.

### 5.1. Detection results

For detection, we fine-tune the decoder of the LDM with a random key $m$, generate 1000 images and use the test of Eq. (1). We report the tradeoff between True Positive Rate (TPR), *i.e.* the probability of flagging a generated image and the FPR, while varying $\tau \in \{0, .., 48\}$. For instance, for $\tau = 0$, we flag all images so FPR $= 1$, and TPR $= 1$. The TPR is measured directly, while the FPR is inferred from Eq. (2), because it would otherwise be too small to be measured on reasonably sized problems (this approximation is validated experimentally in App. B.6). The experiment is run on 10 random signatures and we report averaged results.

Figure 3 shows the tradeoff under image transformations. For example, when the generated images are not modified, Stable Signature detects 99% of them, while only 1 vanilla image out of $10^9$ is flagged. At the same FPR $= 10^{-9}$, Stable Signature detects 84% of generated images for a crop that keeps 10% of the image, and 65% for a transformation that combines a crop, a color shift, and a JPEG compression. For comparison, we report results of a state-of-the-art passive method [12], applied on resized and

[2]The transformation needs to be differentiable in pixel space. This is not the case for JPEG compression so we use the forward attack simulation layer introduced by Zhang *et al*. [101].

[3]We refrain from experimenting with pre-existing third-party generative models, such as Stable Diffusion or LDMs, and instead use a large diffusion model (2.2B parameters) trained on an internal dataset of 330M licensed image-text pairs.

compressed images. As expected, we observe that these baseline results have orders of magnitudes larger FPR than Stable Signature, which actively marks the content.

### 5.2. Identification results

Each Bob has its own copy of the generative model. Given an image, the goal is to find if any of the $N$ Bobs created it (detection) and if so, which one (identification). There are 3 types of error: *false positive*: flag a vanilla image; *false negative*: miss a generated image; *false accusation*: flag a generated image but identify the wrong Bob.

For evaluation, we fine-tune $N' = 1000$ models with random signatures. Each model generates 100 images. For each of these 100k watermarked images, we extract the Stable Signature message, compute the matching score with all $N$ signatures and select the Bob with the highest score. The image is predicted to be generated by that Bob if this score is above threshold $\tau$. We determined $\tau$ such that $\text{FPR} = 10^{-6}$, see Eq. (3). For example, for $N = 1$, $\tau = 41$ and for $N = 1000$, $\tau = 44$. Accuracy is extrapolated beyond the $N'$ Bobs by adding additional signatures and having $N > N'$ (*e.g.* Bobs that have not generated any images).

Figure 4 reports the per-transformation identification accuracy. For example, we identify a Bob among $N$=$10^5$ with 98% accuracy when the image is not modified. Note that for the combined edit, this becomes 40%. This may still be dissuasive: if a Bob generates 3 images, he will be identified 80% of the time. We observe that at this scale, the false accusation rate is zero, *i.e.* we never identify the wrong Bob. This is because $\tau$ is set high to avoid FPs, which also makes false accusations unlikely. We observe that the identification accuracy decreases when $N$ increases, because the threshold $\tau$ required to avoid false positives is higher when $N$ increases, as pointed out by the approximation in (3). In a nutshell, by distributing more models, Alice trades some accuracy of detection against the ability to identify Bobs.

## 6. Experimental Results

The previous section showed how to leverage watermarks for detection and identification of images generated from text prompts. We now present more general results on robustness and image quality for different generative tasks. We also compare Stable Signature to post-generation watermarking algorithms.

### 6.1. Tasks & evaluation metrics

Since our method only involves the LDM decoder, it is compatible with many generative tasks. We evaluate text-to-image generation and image edition on the validation set of MS-COCO [47], super-resolution and inpainting on the validation set of ImageNet [16] (all evaluation details are available in App. A.3).

We evaluate the image distortion with the Peak Signal-to-Noise Ratio (PSNR), which is defined as $\text{PSNR}(x, x') = -10 \cdot \log_{10}(\text{MSE}(x, x'))$, for $x, x' \in [0, 1]^{c \times h \times w}$, as well as Structural Similarity score (SSIM) [90]. They compare images generated with and without watermark. On the other hand, we evaluate the diversity and quality of the generated images with the Fréchet Inception Distance (FID) [34]. The bit accuracy – the percentage of bits correctly decoded – evaluates the watermarks' robustness.

### 6.2. Image generation quality

Figure 6 shows qualitative examples of how the image generation is altered by the latent decoder's fine-tuning. The difference is very hard to perceive even for a trained eye. This is surprising for such a low PSNR, especially since the watermark embedding is not constrained by any Human Visual System like in professional watermarking techniques. Most interestingly, the LDM decoder has indeed learned to add the watermark signal only over textured areas where the human eyes are not sensitive, while the uniform backgrounds are kept intact (see the pixel-wise difference).

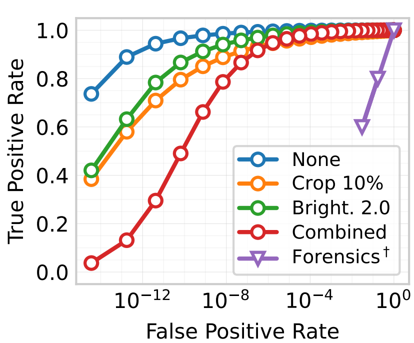



Figure 3. **Detection results**. TPR/FPR curve of the detection under different transformations. Forensics† indicates passive detection (without watermark) [12].

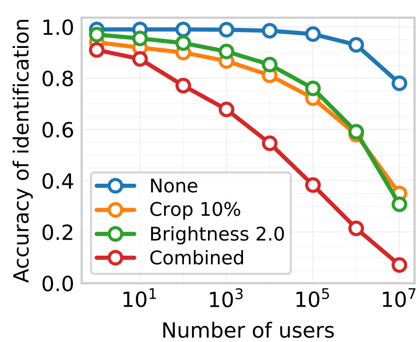



Figure 4. **Identification results**. Proportion of well-identified Bobs. Detection with FPR=$10^{-6}$ is run beforehand, and we consider it an error if the image is not flagged.


Figure 5. Transformations evaluated in Sec. 5 & 6. 'Combined' is made of crop 50%, brightness adjustment 1.5 and JPEG 80 compression.

| | | | PSNR / SSIM ↑ | FID ↓ | Bit accuracy ↑ on: None | Crop | Brigh. | Comb. |
|---|---|---|---|---|---|---|---|---|
| Tasks | Text-to-Image | LDM [68] | 30.0 / 0.89 | 19.6 (−0.3) | 0.99 | 0.95 | 0.97 | 0.92 |
| | Image Edition | DiffEdit [13] | 31.2 / 0.92 | 15.0 (−0.3) | 0.99 | 0.95 | 0.98 | 0.94 |
| | Inpainting - Full | Glide [57] | 31.1 / 0.91 | 16.8 (+0.6) | 0.99 | 0.97 | 0.98 | 0.93 |
| | - Mask only | | 37.8 / 0.98 | 9.0 (+0.1) | 0.89 | 0.76 | 0.84 | 0.78 |
| | Super-Resolution | LDM [68] | 34.0 / 0.94 | 11.6 (+0.0) | 0.98 | 0.93 | 0.96 | 0.92 |
| WM Methods | *Post generation* | | | | | | | |
| | Dct-Dwt [14] | 0.14 (s/img) | 39.5 / 0.97 | 19.5 (−0.4) | 0.86 | 0.52 | 0.51 | 0.51 |
| | SSL Watermark [25] | 0.45 (s/img) | 31.1 / 0.86 | 20.6 (+0.7) | 1.00 | 0.73 | 0.93 | 0.66 |
| | FNNS [42] | 0.28 (s/img) | 32.1 / 0.90 | 19.0 (−0.9) | 0.93 | 0.93 | 0.91 | 0.93 |
| | HiDDeN [108] | 0.11 (s/img) | 32.0 / 0.88 | 19.7 (−0.2) | 0.99 | 0.97 | 0.99 | 0.98 |
| | *Merged in generation* | | | | | | | |
| | Stable Signature | 0.00 (s/img) | 30.0 / 0.89 | 19.6 (−0.3) | 0.99 | 0.95 | 0.97 | 0.92 |

Table 1. Generation quality and comparison to post-hoc watermarking on 512×512 images and 48-bit signatures. PSNR and SSIM are computed between generations of the original and watermarked generators. For FID, we show in (color) the difference with regards to original. Post-hoc watermarking is evaluated on text-generated images. (App. B.2 gives results on more transformations, and App. A gives more details on the evaluations.) Overall, Stable Signature has minimal impact on generation quality. It has comparable robustness to post-hoc methods while being rooted in the generation itself.

Table 1 presents a quantitative evaluation of image generation on the different tasks. We report the FID, and the average PSNR and SSIM that are computed between the images generated by the fine-tuned LDM and the original one. The results show that no matter the task, the watermarking has very small impact on the FID of the generation.

The average PSNR is around 30 dB and SSIM around 0.9 between images generated by the original and a watermarked model. This is a bit low from a watermarking perspective, because we do not explicitly optimize for them. However, in a real world scenario, one would only have the watermarked version of the image. Therefore it is not as important to be extremely close to the original image, we just need to generate artifacts-free images. Without access to the image generated by the original LDM, it is very hard to tell whether a watermark is present or not.

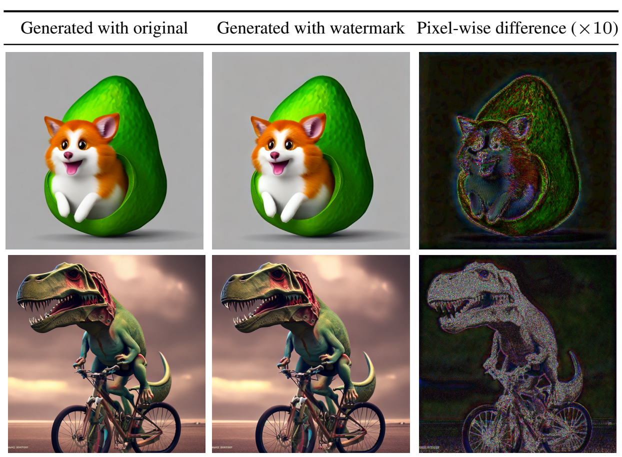



Figure 6. Images generated with Stable Diffusion. The PSNR is 35.4 dB in the first row and 28.6 dB in the second. Images generated with Stable Signature look natural because modified areas are located where the eye is not sensitive. More examples in App. C.

Table 2. Watermark robustness on image transformations applied before decoding, details of which are available in App. A.3. We report the bit accuracy, averaged over 10 × 1k images generated from COCO prompts with 10 different keys.

| **Attack** | **Bit acc.** | | | | |
|---|---|---|---|---|---|
| None | 0.99 | Comb. | 0.92 | Sharpness 2.0 | 0.99 |
| Crop 0.1 | 0.95 | Bright. 2.0 | 0.97 | Med. Filter $k$=7 | 0.94 |
| JPEG 50 | 0.88 | Cont. 2.0 | 0.98 | Resize 0.7 | 0.91 |
| | | Sat. 2.0 | 0.99 | Text overlay | 0.99 |

## 6.3. Watermark robustness

We evaluate the robustness of the watermark to different image transformations applied before extraction. For each task, we generate 1k images with 10 models fine-tuned for different messages, and report the average bit accuracy in Table 1. Additionally, Table 2 reports results on more image transformations for images generated from COCO prompts. The main transformations are presented in Fig. 5 (more evaluation details are available in App. A.3).

We see that the watermark is indeed robust for several tasks and across transformations. The bit accuracy is always above 0.9, except for inpainting, when replacing only the masked region of the image (between $1-50\%$ of the image, with an average of 27% across masks). Besides, the bit accuracy is not perfect even without edition, mainly because there are images that are hard to watermark (*e.g.* the ones that are very uniform, like the background in Fig. 6) and for which the accuracy is lower.

Note that the robustness comes even without any transformation during the LDM fine-tuning phase: it is due to the watermark extractor. If the watermark embedding pipeline is learned to be robust against an augmentation, then the LDM learns how to produce watermarks that are robust against it during fine-tuning.

## 6.4. Comparison to post-hoc watermarking

An alternative way to watermark generated images is to process them after the generation (post-hoc). This may

Table 3. Quality-robustness trade-off during fine-tuning.

| $\lambda_i$ for fine-tuning | 0.8 | 0.4 | 0.2 | 0.1 | 0.05 | 0.025 |
|---|---|---|---|---|---|---|
| PSNR ↑ | 31.4 | 30.6 | 29.7 | 28.5 | 26.8 | 24.6 |
| Bit acc. ↑ on 'comb.' | 0.85 | 0.88 | 0.90 | 0.92 | 0.94 | 0.95 |

Table 4. Role of the attack simulation layer at pre-training.

| Seen at $\mathcal{W}$ training | Bit accuracy ↑ at test time: | | | | |
|---|---|---|---|---|---|
| | Crop 0.1 | Rot. 90 | JPEG 50 | Bright. 2.0 | Res. 0.7 |
| ✗ | 1.00 | 0.56 | 0.50 | 0.99 | 0.48 |
| ✔ | 1.00 | 0.99 | 0.90 | 0.99 | 0.91 |

be simpler, but less secure and efficient than Stable Signature. We compare our method to a frequency based method, DCT-DWT [14], iterative approaches (SSL Watermark [25] and FNNS [42]), and an encoder/decoder one like HiDDeN [108]. We choose DCT-DWT since it is employed by the original open source release of Stable Diffusion [70], and the other methods because of their performance and their ability to handle arbitrary image sizes and number of bits. We use our implementations (see details in App. A.4).

Table 1 compares the generation quality and the robustness over 5k generated images. Overall, Stable Signature achieves comparable results in terms of robustness. HiDDeN's performance is a bit higher but its output bits are not i.i.d. meaning that it cannot be used with the same guarantees as the other methods. We also observe that post-hoc generation gives worse qualitative results, images tend to present artifacts (see Fig. 13 in the supplement). One explanation is that Stable Signature is merged into the high-quality generation process with the LDM auto-encoder model, which is able to modify images in a more subtle way.

### 6.5. Can we trade image quality for robustness?

We can choose to maximize the image quality or the robustness of the watermark thanks to the weight $\lambda_i$ of the perceptual loss in (4). We report the average PSNR of 1k generated images, as well as the bit accuracy obtained on the extracted message for the 'Combined' editing applied before detection (qualitative results are in App. B.1). A higher $\lambda_i$ leads to an image closer to the original one, but to lower bit accuracies on the extracted message, see Table 3.

### 6.6. Attack simulation layer

Watermark robustness against image transformations depends solely on the watermark extractor. here, we pre-train them with or without specific transformations in the simulation layer, on a shorter schedule of 50 epochs, with $128 \times 128$ images and 16-bits messages. From there, we plug them in the LDM fine-tuning stage and we generate 1k images from text prompts. We report the bit accuracy of the extracted watermarks in Table 4. The extractor is naturally robust to some transformations, such as crops or brightness, without being trained with them, while others, like rotations or JPEG, require simulation during training for the watermark to be recovered at test time. Empirically we observed that adding a transformation improves results for the latter, but makes training more challenging.

## 7. Attacks on Stable Signature's Watermarks

We examine the watermark's resistance to intentional tampering, as opposed to distortions that happen without bad intentions like crops or compression (discussed in Sec. 5). We consider two threat models: one is typical for many image watermarking methods [14] and operates at the image level, and another targets the generative model level. For image-level attacks, we evaluate on 5k images generated from COCO prompts. Full details on the following experiments can be found in Appendix A.5.

### 7.1. Image-level attacks

**Watermark removal.** Bob alters the image to remove the watermark with deep learning techniques, like methods used for adversarial purification [78, 94] or neural autoencoders [1, 48]. Note that this kind of attacks has not been explored in the image watermarking literature to our knowledge. Figure 7 evaluates the robustness of the watermark against neural auto-encoders [4, 11, 20, 68] at different compression rates. To reduce the bit accuracy closer to random (50%), the image distortion needs to be strong (PSNR<26). However, assuming the attack is *informed on the generative model*, *i.e.* the auto-encoder is the same as the one used to generate the images, the attack becomes much more effective. It erases the watermark while achieving high quality (PSNR>29). This is because the image is modified precisely in the bandwidth where the watermark

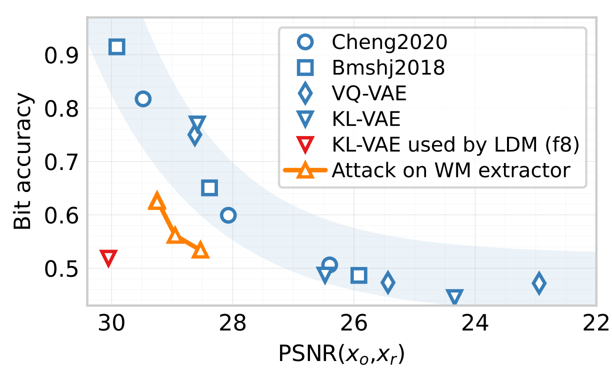



Figure 7. **Removal attacks.** $x_o$ is the image produced by the original generator, $x_r$ is the version produced by the watermarked generator and then attacked. Bit accuracy is on the watermark extracted from $x_r$. Neural auto-encoders [4, 11, 20] follow the same trend , except for the one used by LDM ('KL-f8' for our LDM). When access to the watermark extractor is granted, adversarial attacks also remove the watermark at lower PSNR budget.

is embedded. Note that this assumption is strong, because Alice does not need to distribute the original generator.

**Watermark removal & embedding (white-box).** To go further, we assume that the attack is *informed on the watermark extractor* – *e.g*. because it has leaked. Bob can use an adversarial attack to remove the watermark by optimizing the image under a PSNR constraint. The objective is to minimize the $\ell_2$ distance between a random binary message sampled beforehand and the extractor's output, effectively replacing the original signature with a random one. It makes it possible to erase the watermark with a lower distortion budget, as seen in Fig. 7.

Instead of removing the watermark, an attacker could embed a signature into vanilla images (unauthorized embedding [14]) to impersonate another Bob of whom they have a generated image. It highlights the importance of keeping the watermark extractor private.

### 7.2. Network-level attacks

**Model purification.** Bob gets Alice's generative model and uses a fine-tuning process akin to Sec. 4.2 to eliminate the watermark embedding – that we coin *model purification*. This involves removing the message loss $\mathcal{L}_\mathrm{m}$, and shifting the focus to the perceptual loss $\mathcal{L}_\mathrm{i}$ between the original image and the one reconstructed by the LDM auto-encoder.

Figure 8 shows the results of this attack for the MSE loss. The PSNR between the watermarked and purified images is plotted at various stages of fine-tuning. Empirically, it is difficult to significantly reduce the bit accuracy without compromising the image quality: artifacts start to appear during the purification.

**Model collusion.** Users may collude by aggregating their models. For instance, Bob$^{(i)}$ and Bob$^{(j)}$ can average the weights of their models (like Model soups [91]) creating a new model to deceive identification. We found that the bit at position $\ell$ output by the extractor will be 0 (resp. 1) when the $\ell$-th bits of Bob$^{(i)}$ and Bob$^{(j)}$ are both 0 (resp. 1), and that the extracted bit is random when their bits disagree. We

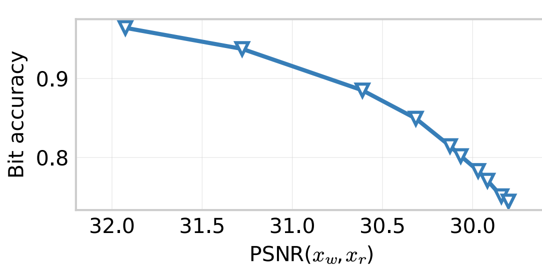
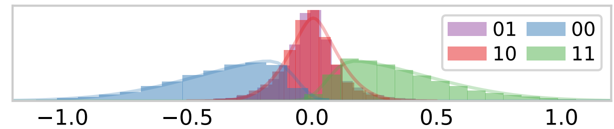



Figure 8. **Robustness to model purification**, *i.e*. fine-tuning the model to remove watermarks. $x_w$ is the watermarked image, $x_r$ is generated with the purified model at different steps of the process.

show the distributions of the soft bits (before thresholding) output by the watermark extractor on images generated by the average model. The $\ell$-th output is labeled by bits of Bob$^{(i)}$ and Bob$^{(j)}$ (`00` means both have `0` at position $\ell$):




This so-called *marking assumption* plays a crucial role in traitor tracing literature [27, 55, 83]. Surprisingly, it holds even though our watermarking process is not explicitly designed for it. The study has room for improvement, such as creating user identifiers with more powerful traitor tracing codes [83] and using more powerful traitor accusation algorithms [27, 55]. Importantly, we found the precedent remarks also hold if the colluders operate at the image level.

## 8. Conclusion & Discussion

By a quick fine-tuning of the decoder of Latent Diffusion Models, we can embed watermarks in all the images they generate. This does not alter the diffusion process, making it compatible with most of LDM-based generative models. These watermarks are robust, invisible to the human eye and can be employed to *detect* generated images and *identify* the user that generated it, with very high performance.

The public release of image generative models has an important societal impact. With this work, we put to light the usefulness of using watermarking instead of relying on passive detection methods. We hope it will encourage researchers and practitioners to employ similar approaches before making their models publicly available.

**Reproducibility Statement.** Although the diffusion-based generative model has been trained on an internal dataset of licensed images, we use the KL auto-encoder from LDM [68] with compression factor $f = 8$. This is the one used by open-source alternatives. Code is available at github.com/facebookresearch/stable_signature.

**Environmental Impact.** We do not expect any environmental impact specific from this work. The cost of the experiments and the method is high, though order of magnitudes less than other computer vision fields. We roughly estimated that the total GPU-days used for running all our experiments to 2000, or $\approx$ 50000 GPU-hours. This amounts to total emissions in the order of 10 tons of $CO_2$eq. This is excluding the training of the generative model itself, since we did not perform that training. Estimations are conducted using the Machine Learning Impact calculator presented by Lacoste *et al.* [43]. We do not consider in this approximation: memory storage, CPU-hours, production cost of GPUs/ CPUs, etc.

## References


[1] Sahar Abdelnabi and Mario Fritz. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 121–140. IEEE, 2021. 7

[2] Mahdi Ahmadi, Alireza Norouzi, Nader Karimi, Shadrokh Samavi, and Ali Emami. Redmark: Framework for residual diffusion watermarking based on deep networks. *Expert Systems with Applications*, 146:113157, 2020. 2, 4

[3] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 2

[4] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018. 7, 14

[5] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022. 17

[6] Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020. 14

[7] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022. 1, 2

[8] Miles Brundage, Shahar Avin, Jack Clark, Helen Toner, Peter Eckersley, Ben Garfinkel, Allan Dafoe, Paul Scharre, Thomas Zeitzoff, Bobby Filar, et al. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. *arXiv preprint arXiv:1802.07228*, 2018. 1

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*. IEEE, 2021. 14

[10] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *European conference on computer vision*, pages 103–120. Springer, 2020. 2

[11] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7939–7948, 2020. 7, 14

[12] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. *arXiv preprint arXiv:2211.00680*, 2022. 2, 4, 5

[13] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022. 1, 2, 6, 13, 16

[14] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital watermarking and steganography*. Morgan kaufmann, 2007. 2, 6, 7, 8

[15] Steffen Czolbe, Oswin Krause, Ingemar Cox, and Christian Igel. A loss function for generative neural networks based on watson's perceptual model. *Advances in Neural Information Processing Systems*, 33:2051–2061, 2020. 4, 15

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5, 14

[17] Emily Denton. Ethical considerations of generative ai. In *AI for Content Creation Workshop CVPR*. IEEE, 2021. 1

[18] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 2

[19] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021. 2

[20] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 2, 7, 14

[21] Hany Farid. Lighting (in) consistency of paint by text. *arXiv preprint arXiv:2207.13744*, 2022. 2

[22] Hany Farid. Perspective (in) consistency of paint by text. *arXiv preprint arXiv:2206.14617*, 2022. 2

[23] Jianwei Fei, Zhihua Xia, Benedetta Tondi, and Mauro Barni. Supervised gan watermarking for intellectual property protection. In *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2022. 2

[24] Pierre Fernandez, Matthijs Douze, Hervé Jégou, and Teddy Furon. Active image indexing. *arXiv preprint arXiv:2210.10620*, 2022. 14

[25] Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. Watermarking images in self-supervised latent spaces. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022. 2, 6, 7, 14, 15

[26] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, pages 3247–3258. PMLR, 2020. 2

[27] Teddy Furon, Arnaud Guyader, and Frédéric Cérou. Decoding Fingerprinting Using the Markov Chain Monte Carlo Method. In *WIFS - IEEE Workshop on Information Forensics and Security*, Tenerife, Spain, Dec. 2012. IEEE. 8

[28] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 1

[29] Matthew Gault. An ai-generated artwork won first place at a state fair fine arts competition, and artists are pissed. *Vice*,

2022. https://www.vice.com/en/article/bvmvqm/an-ai-generated-artwork-won-first-place-at-a-state-fair-fine-arts-competition-and-artists-are-pissed (retrieved 2022-12-14). 1

[30] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2014. 14

[31] Diego Gragnaniello, Davide Cozzolino, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva. Are gan generated images easy to detect? a critical analysis of the state-of-the-art. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2021. 2

[32] Luca Guarnera, Oliver Giudice, and Sebastiano Battiato. Deepfake detection by analyzing convolutional traces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 666–667, 2020. 2

[33] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 2

[34] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 2, 5

[35] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 2

[36] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2

[37] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020. 2

[38] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[39] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 2

[40] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276*, 2022. 2

[41] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 15

[42] Varsha Kishore, Xiangyu Chen, Yan Wang, Boyi Li, and Kilian Q Weinberger. Fixed neural network steganography: Train the images, not the network. In *International Conference on Learning Representations*, 2022. 2, 6, 7, 14

[43] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019. 8

[44] Jae-Eun Lee, Young-Ho Seo, and Dong-Wook Kim. Convolutional neural network-based digital image watermarking adaptive to the resolution of image and watermark. *Applied Sciences*, 2020. 2

[45] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656*, 2018. 2

[46] Dongdong Lin, Benedetta Tondi, Bin Li, and Mauro Barni. Cycleganwm: A cyclegan watermarking method for ownership verification. *arXiv preprint arXiv:2211.13737*, 2022. 1, 2, 3

[47] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4, 5, 13

[48] Wenqing Liu, Miaojing Shi, Teddy Furon, and Li Li. Defending adversarial examples via dnn bottleneck reinforcement. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1930–1938, 2020. 7

[49] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018. 4

[50] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 2

[51] Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. Distortion agnostic deep watermarking. In *CVPR*. IEEE, 2020. 2

[52] Jingwei Ma, Lucy Chai, Minyoung Huh, Tongzhou Wang, Ser-Nam Lim, Phillip Isola, and Antonio Torralba. Totems: Physical objects for verifying visual integrity. In *European Conference on Computer Vision*, pages 164–180. Springer, 2022. 2

[53] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *International Conference on Multimedia*. ACM, 2010. 13

[54] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do gans leave artificial fingerprints? In *2019 IEEE conference on multimedia information processing and retrieval (MIPR)*, pages 506–511. IEEE, 2019. 2

[55] Peter Meerwald and Teddy Furon. Towards practical joint decoding of binary Tardos fingerprinting codes. *IEEE Transactions on Information Forensics and Security*, 7(4):1168–1180, Apr. 2012. 8

[56] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022. 2

[57] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and

Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 6, 13, 16

[58] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 2

[59] Guangyu Nie, Changhoon Kim, Yezhou Yang, and Yi Ren. Attributing image generative models using latent fingerprints. *arXiv preprint arXiv:2304.09752*, 2023. 2

[60] Zoe Papakipos and Joanna Bitton. Augly: Data augmentations for robustness. *arXiv preprint arXiv:2201.06494*, 2022. 13

[61] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems 32, 2019. 16

[62] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 2

[63] Tong Qiao, Yuyan Ma, Ning Zheng, Hanzhou Wu, Yanli Chen, Ming Xu, and Xiangyang Luo. A novel model watermarking for protecting generative adversarial network. *Computers & Security*, 127:103102, 2023. 2

[64] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1

[65] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2, 13

[66] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 2, 13

[67] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 2

[68] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 2, 4, 6, 7, 8, 14, 16

[69] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2, 13

[70] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. Stable diffusion. Accessed Nov. 30 2022, 2022. 1, 7

[71] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022. 1, 2

[72] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. *ICML*, 2019. 17

[73] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 2

[74] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 2, 13

[75] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 14

[76] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 2

[77] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. Defake: Detection and attribution of fake images generated by text-to-image diffusion models. *arXiv preprint arXiv:2210.06998*, 2022. 2

[78] Changhao Shi, Chester Holtz, and Gal Mishne. Online adversarial purification based on self-supervision. *arXiv preprint arXiv:2101.09387*, 2021. 7

[79] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2

[80] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[81] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ICLR*, 2021. 13

[82] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2149–2159, 2022. 13, 20

[83] Gábor Tardos. Optimal probabilistic fingerprint codes. *Journal of the ACM (JACM)*, 55(2):1–24, 2008. 8

[84] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on international conference on multimedia retrieval*, pages 269–277, 2017. 2

[85] Dani Valevski, Matan Kalman, Yossi Matias, and Yaniv Leviathan. Unitune: Text-driven image editing by fine tuning an image generation model on a single image. *arXiv preprint arXiv:2210.09477*, 2022. 2
[86] Vedran Vukotić, Vivien Chappelier, and Teddy Furon. Are deep neural networks good for blind image watermarking? In *WIFS*, 2018. 2
[87] Steven Walton, Ali Hassani, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Stylenat: Giving each head a new perspective. *arXiv preprint arXiv:2211.05770*, 2022. 2
[88] Sheng-Yu Wang, Oliver Wang, Andrew Owens, Richard Zhang, and Alexei A Efros. Detecting photoshopped faces by scripting photoshop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10072–10081, 2019. 2
[89] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8695–8704, 2020. 2
[90] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5
[91] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022. 8
[92] Chen Henry Wu and Fernando De la Torre. Unifying diffusion models' latent space, with applications to cyclediffusion and guidance. *arXiv preprint arXiv:2210.05559*, 2022. 2
[93] Hanzhou Wu, Gen Liu, Yuwei Yao, and Xinpeng Zhang. Watermarking neural networks with watermarked images. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(7):2591–2601, 2020. 2
[94] Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models. In *International Conference on Machine Learning*, pages 12062–12072. PMLR, 2021. 7
[95] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. In *ICLR*, 2020. 13
[96] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021. 2
[97] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7556–7566, 2019. 2
[98] Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14448–14457, 2021. 2, 3
[99] Ning Yu, Vladislav Skripniuk, Dingfan Chen, Larry Davis, and Mario Fritz. Responsible disclosure of generative models using scalable fingerprinting. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 2
[100] Chaoning Zhang, Philipp Benz, Adil Karjauv, Geng Sun, and In So Kweon. Udh: Universal deep hiding for steganography, watermarking, and light field messaging. *NeurIPS*, 2020. 2
[101] Chaoning Zhang, Adil Karjauv, Philipp Benz, and In So Kweon. Towards robust deep hiding under non-differentiable distortions for practical blind watermarking. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5158–5166, 2021. 4, 13
[102] Jie Zhang, Dongdong Chen, Jing Liao, Han Fang, Weiming Zhang, Wenbo Zhou, Hao Cui, and Nenghai Yu. Model watermarking for image processing networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12805–12812, 2020. 2
[103] Kevin Alex Zhang, Alfredo Cuesta-Infante, Lei Xu, and Kalyan Veeramachaneni. Steganogan: High capacity image steganography with gans. *arXiv preprint arXiv:1901.03892*, 2019. 14
[104] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 1, 2
[105] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*. IEEE, 2018. 4, 15
[106] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *2019 IEEE international workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2019. 2
[107] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2185–2194, 2021. 2
[108] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *ECCV*, 2018. 2, 3, 4, 6, 7, 13