

LaWa: Using Latent Space for In-Generation Image Watermarking

Ahmad Rezaei^{1*}, Mohammad Akbari², Saeed Ranjbar Alvar², Arezou Fatemi²,
and Yong Zhang²

¹ University of British Columbia
ahnr@mail.ubc.ca

² Huawei Technologies Canada Co. Ltd.
{mohammad.akbari,saeed.ranjbar.alvar1,yong.zhang3}@huawei.com

Abstract. With generative models producing high quality images that are indistinguishable from real ones, there is growing concern regarding the malicious usage of AI-generated images. Imperceptible image watermarking is one viable solution towards such concerns. Prior watermarking methods map the image to a latent space for adding the watermark. Moreover, Latent Diffusion Models (LDM) generate the image in the latent space of a pre-trained autoencoder. We argue that this latent space can be used to integrate watermarking into the generation process. To this end, we present LaWa, an in-generation image watermarking method designed for LDMs. By using coarse-to-fine watermark embedding modules, LaWa modifies the latent space of pre-trained autoencoders and achieves high robustness against a wide range of image transformations while preserving perceptual quality of the image. We show that LaWa can also be used as a general image watermarking method. Through extensive experiments, we demonstrate that LaWa outperforms previous works in perceptual quality, robustness against attacks, and computational complexity, while having very low false positive rate. Code is available here.

Keywords: Image Watermarking · Responsible AI · Image Generation

1 Introduction

With rapid advancements in generative models, AI-generated content (AIGC) in different modalities including images [35, 38–40], video [21, 45], text [4, 9, 48], and 3D [29, 37] can be generated with high quality. Text-to-image diffusion models such as Stable Diffusion [39] and DALL·E 2 [38] are open to public and have shown stunning performance in generation of photo-realistic images that are indistinguishable from real ones. Such tools can be misused in different ways such as faking AI-generated images as human-created artworks, generating fake news, impersonation, and copyright infringement [5, 50, 67]. Such threats raise concerns about our confidence and trust in the authenticity of photo-realistic images. Thus, responsible implementation of generative AI services needs to be

* Work done during an internship at Huawei Technologies Canada Co. Ltd.

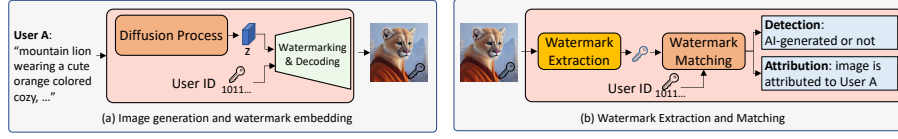


Fig. 1: An overview of in-generation image watermarking for LDMs. (a) generation and watermark (i.e., User ID) embedding. (b) Watermark extraction and matching for detection and attribution.

considered by service providers. Specifically, two problems including detection and attribution should be addressed. In detection, model developers detect if an image is generated by their model, i.e., if the image is AI-generated or not. In attribution, the developers attribute an image to the user who generated the image using their service.

Imperceptible image watermarking is a potential solution for the aforementioned problems. It involves embedding a message in the image without damaging its utility. A watermark extraction module is also required to accurately extract the embedded message from the watermarked image even if the image is modified. In the case of AI-generated images, the extracted message is used to detect such images and attribute them to the responsible user who created them (Fig. 1).

For AIGC images, existing watermarking methods can be applied after the images are generated (called post-generation methods in this paper) [14, 60]. Such post-generation watermarking introduces an overhead on the generation service due to the extra watermark post-processing. The watermarking process generally includes embedding the watermark into a latent feature representation of an image [14, 31, 51, 58]. The latent feature is either learned end-to-end in encoder-decoder-based watermarking methods [31, 58] or it is extracted from the latent space of a pre-trained neural network [14, 51]. In the case of Latent Diffusion Models (LDMs), generated images are in the latent space of a pre-trained autoencoder model. Thus, the latent features corresponding to the generation and watermarking procedures are de-coupled, which is a sub-optimal solution for combined image generation and watermarking problem.

In this paper, we propose LaWa, an in-generation watermarking method that effectively changes the latent feature of pre-trained LDMs to integrate watermarking into the generation process. Thus, generated images already conceal a watermark. Only a few prior works have addressed this problem [13, 53]. Stable Signature [13] is a model watermarking method for the decoder of LDMs. Since it requires fine-tuning a new decoder model for each watermark message, its application for in-generation watermarking has high computational cost and limited scalability for an image generation service. Tree-Ring [53] watermarking modifies the initial noise vector used for image generation. It is only applicable to deterministic sampling [36] and requires image-to-noise inversion process for watermark extraction. Such process for text-guided diffusion does not always yield to the real noise vector [20, 33].

To address these limitations, LaWa incorporates novel coarse-to-fine multi-scale embedding modules into the frozen intermediate layers of the LDM de-

coder to ensure robustness against severe geometrical attacks. LaWa achieves high-payload in-generation watermarking with only one decoder model. Thus, it can handle many users for an image generation service with low computational cost. The focus of LaWa is on black-box watermarking, where users do not have access to the image generation model. In comparison with post-generation watermarking, we show that using the same latent features for image generation and watermarking improves the quality vs. robustness trade-off. Moreover, we show that LaWa can convert a pre-trained autoencoder into a robust general image watermarking method. In summary, our contributions are as follows:

- A multi-scale latent modification mechanism for pre-trained generative autoencoders that is compatible with any autoencoder and is robust to a broad range of image modifications
- A simple yet effective spatial watermark coding that improves the trade-off between robustness and perceptual quality
- An in-generation image watermarking approach that can be used for any pre-trained LDM and any image generation task without further fine-tuning of the LDM
- Extending the proposed in-generation watermarking method to work as a general post-generation image watermarking technique
- Achieving state-of-the-art post- and in-generation results in perceptual quality, robustness, and computational complexity.

2 Related Work

Detection of AI-Generated Images. Considering the risks of AIGC, many works focus on the passive detection of generated/manipulated images. These methods are well-studied for deep-fakes using inconsistencies in generated images [7, 17, 28] as well as generator traces in the spatial [32, 55] or frequency [15, 64] domains. However, they have poor performance because they fall behind the rapid evolution of generative models. Similar approaches are proposed for diffusion models [10, 42], but they are also shown to suffer from low detection accuracy and high false rate [13].

Image Watermarking. There are three main categories of image watermarking methods. Transform-based methods embed the watermark in a spatial [16, 46] or frequency [22, 23, 34] domain, which can achieve great imperceptibility, but have low bit extraction robustness to even minor image modifications. Encoder-decoder-based methods use the encoder to concurrently create a latent feature and add the watermark before recreating the marked image [6, 27, 31, 58, 60, 66]. Despite their good robustness, encoder-decoder networks may not generalize well to images out of the training data distribution and have strong trade-off between payload and utility. The third group uses the latent space of a fixed pre-trained network to add the watermark in several iterations [14, 26, 51, 61], which substantially increases the watermarking time. RoSteALS proposes using latent space of autoencoders for steganography [6], but it does not consider image watermarking attack constraints. The proposed

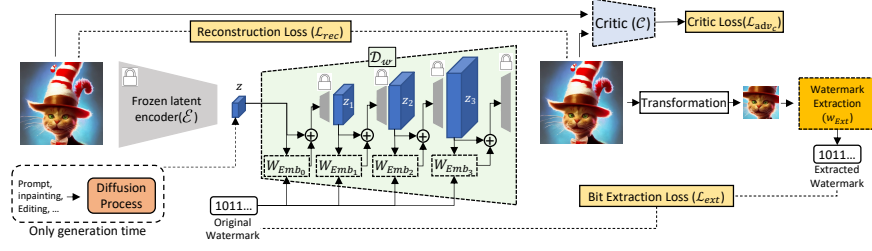


Fig. 2: Overall framework including the procedure of injecting the watermark embedding modules (W_{Emb}) to LDM’s decoder ($\mathcal{D}$) as well as the end-to-end training process. During training, LDM’s frozen encoder is used to generate latent z while at inference time, z is generated by the diffusion process.

solution has no robustness to geometrical image modifications. Such limitation is critically important in the AIGC application scenario as users can easily evade the watermark.

In-Generation Image Watermarking. Previous works on integrating watermarking and image generation are mostly for model watermarking [49]. Some methods watermark the entire training dataset with one message [54, 56, 65] so that the output of the network carries the same message as well. Such methods are not extensible to new messages and require new training for every new message. Specifically for GANs, [57] proposes a solution to merge watermarking into the training procedure of the network. At the inference time, the generated images conceal a fingerprint specific to the user creating the image. This solution requires training the GAN model from scratch and is not applicable to pre-trained models.

In-generation watermarking for diffusion models is a recent research topic in the literature [13, 53]. Tree-Ring watermarking [53] modifies the initial noise vector used in image generation. This method cannot conceal a specific bit string into the image and is only applicable to deterministic sampling [36] in the diffusion process. For watermark extraction, an inversion process should accurately map the watermarked image to the initial noise. [20, 33] show that such inversion for text-guided diffusion does not always yield to the real noise vector, which can damage watermark extraction performance. Stable Signature [13] is a model watermarking method for the image decoder of a pre-trained LDM. If used for in-generation watermarking, a copy of the decoder model should be fine-tuned and stored for each watermark message, substantially increasing the computational cost and limiting the scalability of this solution.

3 Methodology

The overall framework of how LaWa embeds and extracts messages is shown in Fig. 1 (in-generation watermarking) and Fig. 4 (image watermarking). In the embedding phase, a latent is first generated by the diffusion process. Given the watermark message (i.e., user ID), LaWa enables the pre-trained LDM’s latent

decoder to simultaneously watermark and decode the latent, which results in generating the watermarked image. The watermark message can later be extracted using the extraction module to be used for detection and attribution by matching the message with one of existing watermarks.

Figure 2 illustrates the high-level procedure of modifying the LDM’s decoder by injecting watermark embedding modules at intermediate layers of the decoder. The end-to-end training process of the added embedding and extraction networks is also shown, which will be described in details in the following subsections.

3.1 Modified Latent Decoder

In general, diffusion models are trained either in the image space [35, 40] or in a compact latent space [18, 38, 39] for more computational efficiency. For LDMs, the output of the diffusion process is in this latent space. Besides the diffusion process, LDMs also include an image autoencoder. The image encoder, $\mathcal{E}$, is used to downsample the input image $x \in \mathbb{R}^{H \times W \times 3}$ by a factor of f to a latent feature, z , where $z = \mathcal{E}(x) \in \mathbb{R}^{H/f \times W/f \times C}$ [39]. The decoder model, $\mathcal{D}$, then upsamples z to create the reconstructed image $\hat{x} = \mathcal{D}(\mathcal{E}(x))$. At the generation time, the diffusion process generates the latent feature z , which is mapped to the pixel space using the decoder: $\hat{x} = \mathcal{D}(z)$. The upsampling in $\mathcal{D}$ is a multi-step process in which latent size is scaled up by a factor of 2 after each step.

To embed the watermark information, we feed the watermark message into the latent feature before each upsampling step of $\mathcal{D}$. This results in a multi-scale coarse-to-fine process of embedding the watermark into the latent space features. More specifically, assuming a k -bit watermark message $m \in \{0, 1\}^k$, our watermark embedding module at i th upsampling step, W_{Emb_i} , accepts m and the latent z_i as input and generates a residual latent, δz_i , with the same size as z_i : $\delta z_i = W_{Emb_i}(z_i, m)$, where $i \in \{0, \dots, \frac{f}{2} - 1\}$ shows the upsampling step of $\mathcal{D}$. The watermarked latent, z_{w_i} is then constructed as:

$$z_{w_i} = z_i + \delta z_i. \quad (1)$$

As illustrated in Fig. 3, W_{Emb_i} comprises of a linear layer that maps m to a noise block $b_i \in \mathbb{R}^{B \times B \times C_i}$. C_i is the latent channel size at the i th step and B is the height and width size of b_i that is kept constant for all upsampling steps. After b_i is obtained, we perform spatial watermark coding where b_i is repeated along the height and width dimensions to match the size of z_i . As a result of the applied spatial watermark coding, the watermark noise carries repeated watermark message with the idea to create sufficient redundancy in the noise to improve robustness to geometrical attacks while minimally damaging the perceptual quality. The watermark noise is then passed through a convolution layer to generate δz_i . To ensure W_{Emb_i} does not initially change the latent feature, we follow [62] and initialize the weights and biases of the convolution layer to 0.

Unlike the original decoder $\mathcal{D}$ that reconstructs the image without watermark $\hat{x} = \mathcal{D}(z)$, the modified latent decoder $\mathcal{D}_w$ generates the watermarked image $\hat{x}_w = \mathcal{D}_w(z)$.

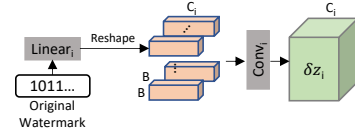


Fig. 3: Structure of the watermark embedding module W_{Emb_i}

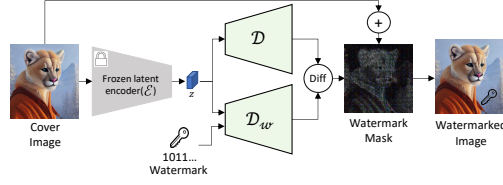


Fig. 4: Diagram of general image watermarking process with LaWa

3.2 Training

In the training process, the LDM’s pre-trained image encoder as well as the original layers of the decoder are kept frozen. This ensures that training the watermark embedding modules do not degrade the image generation quality.

To ensure the perceptual similarity of the watermarked image $\hat{x}_w$ and the original generated image $\hat{x}$, we use a combination of pixel-wise distortion and perceptual loss functions. For the distortion loss, MSE (l_2 distance) loss defined as $\mathcal{L}_I = \|\hat{x}_w - \hat{x}\|^2$ is used. For the perceptual loss function, we employ LPIPS loss [63] to minimize the perceptual distortion between $\hat{x}_w$ and $\hat{x}$.

To further improve the quality of the watermarked image, an adversarial critic network $\mathcal{C}$ using Wasserstein GAN [2] is also implemented. Thus, the corresponding adversarial training includes two loss functions as follows:

$$\mathcal{L}_{adv\mathcal{D}_w} = -\mathbb{E}_{\hat{x}_w \sim P_{\hat{x}_w}}[\mathcal{C}(\hat{x}_w)], \quad (2)$$

$$\mathcal{L}_{adv\mathcal{C}} = \mathbb{E}_{\hat{x}_w \sim P_{\hat{x}_w}}[\mathcal{C}(\hat{x}_w)] - \mathbb{E}_{\hat{x} \sim P_{\hat{x}}}[\mathcal{C}(\hat{x})], \quad (3)$$

where $\mathcal{L}_{adv\mathcal{C}}$ and $\mathcal{L}_{adv\mathcal{D}_w}$ are the adversarial losses for the critic $\mathcal{C}$ and the modified decoder $\mathcal{D}_w$, respectively. $P_{\hat{x}_w}$ and $P_{\hat{x}}$ are the distribution of the watermarked and original images. The overall reconstruction loss is then defined as:

$$\mathcal{L}_{rec} = \lambda_I \mathcal{L}_I + \lambda_{LPIPS} \mathcal{L}_{LPIPS}(\hat{x}_w, \hat{x}) + \lambda_{adv} \mathcal{L}_{adv\mathcal{D}_w}, \quad (4)$$

where λ_I , λ_{LPIPS} , and λ_{adv} are the relative loss weights.

Our watermark extractor, denoted by W_{Ext} , should be robust to different image modifications and attacks. As a result, after the watermarked image is generated, a transformation T is randomly sampled from a set of differentiable transformations, $\mathcal{T}$, including common image processing attacks (see supplementary material for more details). For the differentiability issue of JPEG compression, we used the forward simulation attack introduced in [59]. The selected transformation is then applied to the image and the transformed image is passed through the watermark extraction network, which is based on ResNet50 [19] architecture with the last linear layer changed to output a k -bit message. The output extracted message, $\hat{m} = W_{Ext}(T(\hat{x}))$, should match the original embedded message m . We define the extraction loss as the binary cross entropy (BCE) loss between m and $\hat{m}$, $\mathcal{L}_{ext} = \mathcal{L}_{BCE}(m, \hat{m})$. The final watermark embedding and extraction networks are then optimized to minimize the following total loss:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda \mathcal{L}_{ext}, \quad (5)$$

where λ is the loss weight to control the trade-off between extraction accuracy and image reconstruction quality.

3.3 Image Watermarking

Figure 4 shows the general image watermarking process. Corresponding latent feature z of the cover image x is calculated using the frozen encoder, $z = \mathcal{E}(x)$. The feature z is separately processed by $\mathcal{D}$ and $\mathcal{D}_w$ to create a watermark mask that is computed as $\mathcal{D}(z) - \mathcal{D}_w(z)$. Using this watermark mask, the cover image is then watermarked as $\hat{x}_w = x + \mathcal{D}_w(z) - \mathcal{D}(z)$.

3.4 Watermark Matching

Given an image at the watermark extraction time, our watermark extractor W_{Ext} tries to decode the watermark message $\hat{m}$. To address the detection and attribution problems, $\hat{m}$ needs to be matched with one of existing original messages (i.e., m) stored in the database. Following [13, 56] and to increase the matching robustness against image modifications, we use a soft matching method defined as:

$$M(\hat{m}, m) \geq n \quad n \in \{0, \dots, k\}, \quad (6)$$

where $M(\hat{m}, m)$ counts the number of matching bits in $\hat{m}$ and m , and n is the threshold for flagging a match.

For a successful watermark matching, the rate of false detection and attribution should be at an acceptable level. Potential errors include detecting a vanilla image as watermarked (detection FPR), missing a watermarked image (detection false negative rate), and attributing an image to the wrong user (attribution FPR). Having low error rate is an important practical aspect of this problem.

Assuming that extracted bits from vanilla images are i.i.d Bernoulli random variables with parameter 0.5, the theoretical upper bound of FPR is as follows:

$$FPR_{det}(n) = Pr(M(\tilde{m}, m) > n \mid H_0) = \sum_{i=n+1}^k \binom{k}{i} 0.5^k, \quad (7)$$

$$FPR_{att}(N, n) = 1 - (1 - FPR_{det}(n))^N, \quad (8)$$

where FPR_{det} is the detection FPR, FPR_{att} is the attribution FPR, N is the total number of users, $\tilde{m}$ is any random message, and H_0 is the null hypothesis: "any random message, $\tilde{m}$, match with m in more than n bits". Clearly, the more the number of users, the higher the attribution FPR. By setting the FPR_{att} and total number of users to a desired level, we can obtain the minimum required n for watermark matching by using Eqs. (7) and (8) reversely. More information is provided in the supplementary material.

4 Experiments

In this section, the performance of the proposed method is quantitatively and qualitatively evaluated and compared with the previous works. We also study the generality of LaWa for different image generation tasks. A false positive analysis along with an extensive test for the trade-off between capacity, quality, and robustness is also presented. At the end, we will provide an ablation study to verify the effect of each loss term in our objective function.

4.1 Experimental Settings

To train LaWa’s watermark embedding and extraction modules, we utilize 100K images from MIRFlickR dataset [24], where the images are randomly cropped to 256×256 resolution. The training is performed with an AdamW optimizer with the learning rate of $6e-5$, 40 epochs, and the batch size of 8. For all our experiments, we set the weights in Eqs. (4) and (5) to $\lambda_I = 0.1$, $\lambda_{LPIPS} = 1.0$, $\lambda_{adv} = 1.0$, and $\lambda = 2.0$, which are obtained experimentally. For the main experiments, we use the KL-f8 auto-encoder of Stable Diffusion [39] and 48-bit messages to train the LaWa method unless mentioned otherwise.

To numerically evaluate the distortion of the watermarked and original generated images, we use Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [52]. Moreover, we evaluate the perceptual quality of images using LPIPS similarity score [63] and Single Image Frechet Inception Distance (SIFID) [43].

4.2 Comparison Results

We compare the performance of LaWa with previous in-generation and image watermarking methods using AI-generated and natural image datasets. For AI-generated dataset, we consider using LaWa as both in-generation and post-generation watermarking methods. For the baseline of in-generation methods, we used Stable Signature (SS) [13], which is a model watermarking method for in-generation image watermarking by creating different copies of the decoder. Alternatively, image watermarking methods can be used to watermark the generated images in a post-generation manner. We provide the comparison results with different post-generation techniques including encoder-decoder based (HiD-DeN [66], RivaGAN [60], and RoSteALS [6]), optimization-based (SSL [14] and FNNS [26]), and frequency-based (DCT-DWT [11]) methods. For HiDDeN and RivaGAN, 30-bit and 32-bit pre-trained weights in [25] and [44] are respectively used. For RoSteALS, we trained their model for 48- and 32-bit message lengths with the same dataset and attacks as in LaWa. For SSL and FNNS, we used their publicly available codes and their pre-trained modules to obtain the results.

The upper portion of Tab. 1 summarizes LaWa’s results with 32- and 48-bit watermark messages compared to the baselines in terms of the watermarked image quality as well as the bit extraction accuracy after different attacks. The results are obtained by watermarking 1K generated images (with 1K text prompts

Table 1: Comparison results of LaWa with existing post- and in-generation image watermarking methods in terms of quality, embedding time, and robustness against various attacks. *: in-generation methods.

Method (bit#)	Image quality		Emb. time (ms)	Bit accuracy $\uparrow$										Comb.	Ave.	
	PSNR/SSIM $\uparrow$	LPIPS/SIFID $\downarrow$		None	C. Crop 0.1	R. Crop 0.1	Resize 0.7	Rot. 15	Blur	Contr. 2.0	Bright. 2.0	JPEG 70				
AI-generated Images																
DCT-DWT (32) [11]	39.47/0.97	0.03/0.02	139	0.91	0.51	0.51	0.52	0.52	0.51	0.52	0.51	0.51	0.51	0.51	0.55	
Hidden (30) [25]	32.59/0.95	0.03/0.05	17	0.91	0.91	0.91	0.82	0.79	0.76	0.75	0.74	0.53	0.59	0.77		
SSL (32) [14]	33.23/0.89	0.15/0.28	870	1.00	0.74	0.72	0.99	0.99	1.00	0.96	0.95	0.99	0.85	0.92		
RoSteALS (32) [6]	29.31/0.93	0.04/0.06	144	1.00	0.50	0.51	1.00	0.47	1.00	0.88	0.86	1.00	0.50	0.77		
RivaGan (32) [60]	40.53/0.98	0.03/0.04	57	0.99	0.98	0.98	0.87	0.91	0.99	0.81	0.80	0.98	0.93	0.92		
LaWa* (32)	34.25/0.89	0.03/0.01	1	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.99		
LaWa-post-gen (32)	34.28/0.90	0.03/0.02	33	1.00	1.00	0.96	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.99		
SSL (48) [14]	33.25/0.89	0.15/0.27	870	1.00	0.72	0.70	0.99	0.99	1.00	0.94	0.94	0.99	0.82	0.91		
FNNS (48) [26]	36.84/0.97	0.04/0.05	645	1.00	0.98	0.96	0.95	0.75	0.54	0.86	0.85	0.94	0.91	0.87		
RoSteALS (48) [6]	29.30/0.92	0.05/0.07	144	1.00	0.51	0.50	1.00	0.50	1.00	0.88	0.85	1.00	0.49	0.78		
SS* (48) [13]	32.02/0.84	0.05/0.09	0	0.99	0.95	0.93	0.96	0.81	0.78	0.97	0.96	0.92	0.92	0.92		
LaWa* (48)	33.52/0.86	0.04/0.02	1	1.00	0.95	0.91	0.99	0.96	0.99	1.00	1.00	1.00	0.94	0.97		
LaWa-post-gen (48)	33.45/0.87	0.04/0.02	34	1.00	0.94	0.90	0.96	0.97	0.99	1.00	1.00	1.00	0.95	0.97		
CLIC																
DCT-DWT (32) [11]	38.94/0.97	0.02/0.02	230	0.87	0.51	0.51	0.51	0.52	0.50	0.50	0.52	0.50	0.55			
Hidden (30) [25]	33.33/0.96	0.04/0.04	28	0.89	0.89	0.88	0.86	0.76	0.64	0.76	0.74	0.54	0.58	0.75		
SSL (32) [14]	35.03/0.92	0.16/0.19	1261	1.00	0.86	0.80	1.00	1.00	1.00	0.96	0.93	0.98	0.90	0.94		
RoSteALS (32) [6]	28.22/0.91	0.08/0.04	240	1.00	0.50	0.50	1.00	0.51	1.00	0.90	0.87	1.00	0.50	0.78		
RivaGan (32) [60]	42.04/0.98	0.04/0.06	84	1.00	0.98	0.96	0.99	0.95	0.99	0.89	0.87	0.99	0.96	0.96		
LaWa (32)	36.19/0.94	0.05/0.03	32	1.00	1.00	0.97	0.98	1.00	1.00	1.00	0.99	1.00	0.99	0.99		
SSL (48) [14]	35.03/0.92	0.16/0.19	1261	1.00	0.83	0.78	1.00	1.00	1.00	0.96	0.92	0.98	0.88	0.93		
FNNS (48) [26]	35.54/0.96	0.07/0.20	1116	1.00	0.99	0.95	0.94	0.75	0.51	0.85	0.82	0.91	0.89	0.86		
RoSteALS (48) [6]	27.83/0.91	0.08/0.04	240	1.00	0.50	0.50	1.00	0.48	1.00	0.82	0.80	1.00	0.50	0.76		
LaWa (48)	35.58/0.92	0.05/0.05	34	1.00	1.00	0.97	0.98	0.98	0.98	1.00	0.99	0.98	0.90	0.98		

obtained from MS-COCO [30] and MagicPrompt-SD [41] datasets). Resolution of generated images is set to 512×512 . Each generated image is watermarked with 10 different messages and the average bit accuracy is reported. Combined attack includes a series of 40% area center cropping, brightness 2.0, and JPEG 80. LaWa* and LaWa-post-gen refer to the cases where the proposed method is used as in-generation and post-generation watermarking, respectively.

As shown in Tab. 1, LaWa achieves the best perceptual quality for AI-generated images with 32- and 48-bit versions in LPIPS and SIFID scores. For PSNR and SSIM, LaWa has comparable performance with other methods, while RivaGAN and FNNS outperform the others for 32- and 48-bit watermarks, respectively. For robustness to attacks, LaWa has the best bit accuracy for all attacks for 32-bit watermarks. For 48-bit watermarks, except for 10% random crop attack where FNNS works better (90% bit accuracy of LaWa compared to 96% of FNNS), LaWa has better or comparable bit accuracy. Overall, LaWa outperforms other methods in more number of attacks, specifically for the combined attack. Moreover, LaWa outperforms SS (as an in-generation method) in both image quality and robustness to a variety of attacks. More detailed set of results with different attack parameters is given in the supplementary material. When LaWa is used as a post-generation watermarking method (i.e, decoupled from the generation), the extraction accuracy is nearly unchanged compared to in-generation watermarking.

The computational complexity of all methods compared to ours in terms of averaged embedding time is also given in Tab. 1. Overall, it is seen that

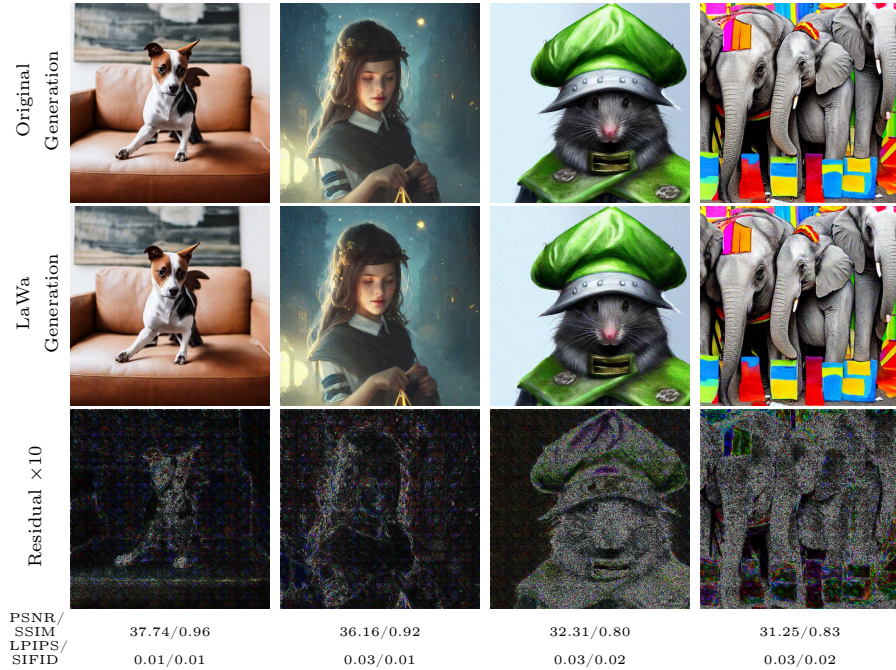


Fig. 5: Qualitative examples of generated images using LaWa, original generated images, and residual images (multiplied by 10). Extensive added noise by LaWa in high frequency textures reduces the PSNR/SSIM without damaging the perceptual quality (LPIPS/SIFID).

post-generation methods result in a significant overhead for watermarking the generated images, especially the iteration-based ones such as SSL and FNNS. On the other hand, the in-generation methods such as SS and ours provide near zero overhead as no post-processing is needed to add the watermark.

We also compare the performance of LaWa in general image watermarking against baseline methods using a natural (i.e., not AI-generated) dataset. Following earlier works in [6, 14], CLIC dataset [47] is used, which has 428 test images with different resolutions. For all images, while maintaining the aspect ratio, larger dimension of each image is resized to 1024 to avoid memory limitation issue for some of the baselines. The results with CLIC dataset are given in the bottom part of Tab. 1. LaWa-post-gen achieves the highest bit accuracy for almost all the attacks and comparable or better visual quality compared to the baselines. Qualitative results for CLIC are in the supplementary material.

4.3 Qualitative Analysis

Fig. 5 shows visual examples of 4 images generated by the original and modified decoders (i.e., original generated vs. watermarked images). The residual images (multiplied by 10) and the numerical quality metrics including PSNR, SSIM, LPIPS, and SIFID are also given for each pair.

Comparing the watermarked and original images, although the calculated PSNR is not very high, it is visually not possible to differentiate them as no visual artifact is evident. This is expected as we trained LaWa focusing on the perceptual quality of the watermarked images while giving lower weight to the MSE loss, which directly affects the PSNR. From the examples in Fig. 5, it is clear that our method has learned to add more noise to the high frequency textures, especially for the generated images with more texture (e.g., the elephants). In practice, when using in-generation watermarking, there is no original image available for comparison as all generated images conceal the watermark. This relaxes the requirement for watermarked images to have a high pixel-wise similarity to the original image. Therefore, we argue that if generated images have no visual artifact, watermark addition is successful even if PSNR is not very high. More qualitative results compared to the previous works are presented in the supplementary material.

4.4 Capacity, Quality, and Robustness Trade-off

Effect of Number of Bits.

We study the effect of the number of bits on generation quality and robustness in Tab. 2. LaWa can hide 64- and 96-bit watermark messages with minor drop in image quality, while achieving near 100% bit accuracy with no attack. With 128-bit messages, minor drop in the quality and bit accuracy against attack is observed compared to embedding shorter messages. Using higher number of bits can further improve the FPR of LaWa for its application in image generation services, which are designed to support a large number of users.

Table 2: Effect of the number of bits on generation quality and bit extraction accuracy.

Bit length	PSNR/ SSIM	LPIPS/ SIFID	None	C. Cro 0.1	Contr. 2.0	Rot. 15	JPEG 70	Comb.
32	34.25/0.89	0.03/0.01	1.00	1.00	1.00	1.00	1.00	0.97
48	33.52/0.86	0.04/0.02	1.00	0.95	1.00	0.96	1.00	0.94
64	33.11/0.83	0.04/0.02	1.00	0.93	0.99	0.92	0.96	0.95
96	32.82/0.83	0.04/0.02	0.99	0.87	0.98	0.77	0.92	0.95
128	31.87/0.82	0.08/0.09	0.95	0.93	0.95	0.84	0.92	0.87

Quality-Robustness Trade-off. We can trade the generation quality for higher bit accuracy by changing λ in Eq. (5). Table 3 presents the performance of LaWa when trained with different λ s. The corresponding PSNR and bit accuracy (after applying combined attack) on 1K images is reported. As observed, by increasing λ , the extraction robustness is improved, but the quality of the watermarked images drops.

4.5 LaWa’s Performance for Different Tasks

In this sub-section, the generality of LaWa for different image generation tasks [39] including text-to-image, inpainting, super-resolution, and image editing is studied. The corresponding performance results are given in Tab. 5. The results are obtained for 1K images generated/watermarked with a separate random watermark message of 48 bits (see the supplementary material for more details). As summarized in the Tab. 5, the achieved bit accuracy averaged over all the tasks

Table 3: Trade-off between image quality and bit accuracy. λ : trade-off weight in Eq. (5).

λ	1.0	1.7	2.0	4.0	7.0	10.0
PSNR $\uparrow$	34.76	33.84	33.52	33.06	30.56	28.47
Bit acc. on comb. $\uparrow$	0.65	0.75	0.94	0.96	0.97	0.99

Table 4: Ablation study on the effect of image quality loss weights.

λ_I	0.1	0.5	1.0
PSNR/SSIM $\uparrow$	33.52/0.86	33.60/0.88	33.94/0.90
LPIPS/SIFID $\downarrow$	0.04/0.02	0.05/0.04	0.06/0.05
Bit acc. on None/comb. $\uparrow$	1.00/0.94	1.00/0.88	0.99/0.84

is higher than 97%, while the perceptual quality is preserved. This shows the applicability and generality of our proposed method to different generation tasks.

4.6 False Positive Analysis

The false positive analysis of LaWa’s performance for both detection and attribution problems is given in this sub-section. As discussed in Sec. 3.4, in the detection problem, we only aim to detect if an image is watermarked while having a low FPR (detecting vanilla images as watermarked). On the other hand, in the attribution problem, we aim to find the user who created the image without making a false attribution to the wrong user (false attribution).

Detection. For detection, we use 1K watermarked images with a single 48-bit message to study the effect of the threshold n used for soft watermark matching on FPR_{det} . For each $n \in \{0, \dots, 48\}$, we use Eq. (7) to calculate FPR_{det} , while TPR_{det} is experimentally obtained for the 1K images under various attacks. We experimentally validate the theoretical FPR_{det} in the supplementary material.

Figure 6a shows LaWa performance at different FPR_{det} . Under no attack, brightness 2.0, and JPEG 60, LaWa successfully detects 99% of the images while only flagging 1 out of 10^{15} vanilla images as watermarked. For more severe attacks such as crop 10% and combined attacks, there is a clear trade-off between detection accuracy and FPR_{det} . LaWa still detects more than 95% of the cropped and 83% of the combined-attacked images while only flagging 1 out of 10^9 vanilla images. We also compare LaWa with Stable Signature (SS) [13] under combined attack and show that LaWa has a much higher TPR.

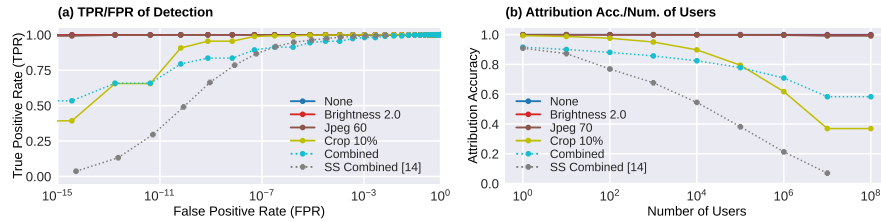


Fig. 6: Watermark matching performance in detection and attribution. (a) Detection TPR/FPR for different number of matching bits under different attacks. (b) Accuracy of attributing the watermarked image to the right user under different attacks and number of users assuming the detection FPR of 10^{-6} .

Table 5: Performance of LaWa in different image generation tasks in terms of quality and robustness.

Task	PSNR/ SSIM	LPIPS/ SIFID	C. Crop 0.1	Contr. 2.0	Rot. 15	JPEG 70
Text-to-Image	33.52/0.86	0.04/0.02	0.95	1.00	0.96	1.00
Inpainting	33.68/0.86	0.04/0.02	0.94	1.00	0.97	0.99
Super-Resolution	35.40/0.89	0.04/0.01	0.96	0.99	0.96	0.98
Image Editing	34.22/0.88	0.04/0.01	0.95	1.00	0.96	0.99

Table 6: Ablation study on the effect of each loss term on image quality and bit extraction robustness.

MSE	LPIPS	Critic	PSNR/SSIM	LPIPS/SIFID	Bit Acc. Comb.
✓			31.26/0.82	0.22/0.52	0.96
	✓		28.83/0.36	0.05/0.41	0.93
✓	✓		29.47/0.47	0.04/0.31	0.85
✓		✓	32.29/0.80	0.10/0.05	0.99
	✓	✓	33.05/0.85	0.05/0.02	0.73
✓	✓	✓	33.52/0.86	0.04/0.02	0.94

Attribution. For the attribution problem, we can use Eqs. (7) and (8) reversely to find the number of possible users for any FPR_{att} . For example, for $FPR_{att} = 10^{-6}$ in Eq. (8), we can calculate the required FPR_{det} for any number of users N . We further use the computed FPR_{det} and Eq. (7) to find the necessary watermark matching threshold n . For example, for $FPR_{att} = 10^{-6}$ and $N = 10^5$, we should use $n = 45$ as the threshold of soft watermark matching for 48-bit watermark messages.

We study LaWa’s attribution accuracy for each N with 50K different 48-bit messages. We generate 2 images per message resulting in a total of 100K images. For $FPR_{att} = 10^{-6}$ and each N , the required threshold n is computed. Using this n , we perform watermark matching between the extracted message from each image and all the other messages. The match with the highest score above n is considered true positive (TP) if it corresponds to the correct user. Accuracy is calculated as $TP/100k$. For N s larger than 50k, we assume some users have not generated any image.

Figure 6b shows the attribution accuracy of our method for different attacks. Under no attack, brightness 2.0, and JPEG 70, LaWa attributes 99% of the generated images to the correct user among 10^6 total users. Accuracy decreases to less than 70% and 60% for more severe combined and 10% crop attacks, respectively. Compared to Stable Signature (SS) [13], LaWa achieves a better performance with 20% accuracy for the combined attack. Moreover, LaWa has zero false attribution when $N = 10^6$, which means no image was attributed to the wrong user. This is expected as the watermark matching threshold is really high at $n = 47$. Overall, as we increase the number of users, the probability of false accusations also increases.

4.7 Learning-based Attacks

We study the impact of learning-based watermark removal attacks, where the watermarked image is passed through an auto-encoder model [1, 3, 8, 12, 39] as an attempt to remove the watermark. Figure 7 shows the robustness of LaWa compared to Stable Signature (SS) [13] against several auto-encoder models including Bmshj2018 [3], Cheng2020 [8], KL-VAE [39], and VQ-VAE [12].

Each type of attack is applied using different quality/downsampling factors used in the auto-encoders. We observe that LaWa is quite robust against the attacks performed by Bmshj2018 with q8, Cheng2020 with q6, KL-VAE with f4, and VQ-VAE with f4. As the attack gets stronger (i.e., images are more

compressed with larger quality or downsampling factors), both the bit accuracy and the PSNR reduce. Meaning if the attack is strong enough to remove the watermark, the quality of the watermarked image will inevitably experience a significant degradation (i.e., $\text{PSNR} < 24$). It is also observed that even when the auto-encoder used in the attack is the same as the one we used to generate/watermark the images (i.e., KL-VAE f8), the attack will not erase the watermark and the bit accuracy remains as high as 85%. In contrast, the bit accuracy of SS [13] drops to 55%. It should be noted that unlike [13], we calculate the PSNR between the watermarked and the attacked images which is a more accurate representation of the attack’s effect.

4.8 Ablation Study

Loss Terms. Table 6 shows the effect of each loss used in Eq. (4). We observe that the Critic network helps to improve the visual quality of the images. Using MSE and Critic losses achieves the best bit accuracy, but this setting leads to low perceptual quality and high visual artifacts in the image. By adding LPIPS loss, such visual artifacts are removed from the generated images.

Loss Weights. Table 4 shows the effect of changing loss weights in Eq. (4). We did this study by fixing both λ_{LPIPS} and λ_{adv} to 1.0 and changing λ_I . We observe that $\lambda_I > 0.1$ provides minor improvement in PSNR/SSIM, but damages LPIPS/SIFID and LaWa’s robustness. More ablation studies over other components of LaWa along with qualitative results are given in the supplementary material.

5 Conclusion

In this paper, we proposed an in-generation image watermarking solution, which is applicable to LDMs and any image generation task. We showed that by injecting the watermark embedding modules into the intermediate layers of the frozen decoder, these modules can learn the decoder-specific mapping of latent space to RGB images. Thus, the watermark message is concealed in the generated images in a way that is imperceptible and robust to image modifications. Our solution is scalable to support a large number of users only with one single decoder model. With the fast evolution of generative AI models and its corresponding risks, model developers are expected to actively help with the safe adaptation of these models. Our work is a step towards responsible AI and we hope it can help model developers to implement their generative AI service while they have countermeasures in place to limit any improper use.

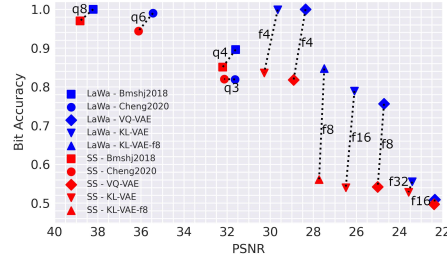


Fig. 7: Effect of the learning-based removal attacks. Results are reported for (SS) [13] and LaWa. q, f : quality/downsampling factors for each attack.

References

1. Akbari, M., Liang, J., Han, J., Tu, C.: Learned bi-resolution image coding using generalized octave convolutions. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 35, pp. 6592–6599 (2021)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan (2017)
3. Ballé, J., Minnen, D., Singh, S., Hwang, S.J., Johnston, N.: Variational image compression with a scale hyperprior (2018)
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. *Advances in neural information processing systems* **33**, 1877–1901 (2020)
5. Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B., Dafoe, A., Scharre, P., Zeitzoff, T., Filar, B., et al.: The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. *arXiv preprint arXiv:1802.07228* (2018)
6. Bui, T., Agarwal, S., Yu, N., Collomosse, J.: Rosteals: Robust steganography using autoencoder latent space. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 933–942 (2023)
7. Chai, L., Bau, D., Lim, S.N., Isola, P.: What makes fake images detectable? understanding properties that generalize. In: *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI* 16. pp. 103–120. Springer (2020)
8. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2020)
9. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311* (2022)
10. Corvi, R., Cozzolino, D., Zingarini, G., Poggi, G., Nagano, K., Verdoliva, L.: On the detection of synthetic images generated by diffusion models. In: *ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 1–5. IEEE (2023)
11. Cox, I., Miller, M., Bloom, J., Fridrich, J., Kalker, T.: Digital watermarking and steganography. Morgan kaufmann (2007)
12. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis (2021)
13. Fernandez, P., Couairon, G., Jégou, H., Douze, M., Furon, T.: The stable signature: Rooting watermarks in latent diffusion models. *ICCV* (2023)
14. Fernandez, P., Sablayrolles, A., Furon, T., Jégou, H., Douze, M.: Watermarking images in self-supervised latent spaces. In: *ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 3054–3058. IEEE (2022)
15. Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D., Holz, T.: Leveraging frequency analysis for deep fake image recognition. In: *International conference on machine learning*. pp. 3247–3258. PMLR (2020)
16. Ghazanfari, K., Ghaemmaghami, S., Khosravi, S.R.: Lsb++: An improvement to lsb+ steganography. In: *TENCON 2011–2011 IEEE Region 10 Conference*. pp. 364–368. IEEE (2011)
17. Gragnaniello, D., Cozzolino, D., Marra, F., Poggi, G., Verdoliva, L.: 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). pp. 1–6. IEEE (2021)
18. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B.: Vector quantized diffusion model for text-to-image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10696–10706 (2022)
 19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
 20. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)
 21. Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., et al.: Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 (2022)
 22. Holub, V., Fridrich, J.: Designing steganographic distortion using directional filters. In: 2012 IEEE International workshop on information forensics and security (WIFS). pp. 234–239. IEEE (2012)
 23. Holub, V., Fridrich, J., Denemark, T.: Universal distortion function for steganography in an arbitrary domain. EURASIP Journal on Information Security **2014**, 1–13 (2014)
 24. Huiskes, M.J., Lew, M.S.: The mir flickr retrieval evaluation. In: MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval. ACM, New York, NY, USA (2008)
 25. Khachatryan, A.: Hidden. <https://github.com/ando-khachatryan/HiDeN> (2019)
 26. Kishore, V., Chen, X., Wang, Y., Li, B., Weinberger, K.Q.: Fixed neural network steganography: Train the images, not the network. In: International Conference on Learning Representations (2021)
 27. Lee, J.E., Seo, Y.H., Kim, D.W.: Convolutional neural network-based digital image watermarking adaptive to the resolution of image and watermark. Applied Sciences **10**(19), 6854 (2020)
 28. Li, Y., Lyu, S.: Exposing deepfake videos by detecting face warping artifacts. arXiv preprint arXiv:1811.00656 (2018)
 29. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 300–309 (2023)
 30. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
 31. Luo, X., Zhan, R., Chang, H., Yang, F., Milanfar, P.: Distortion agnostic deep watermarking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13548–13557 (2020)
 32. Marra, F., Gagnaniello, D., Verdoliva, L., Poggi, G.: Do gans leave artificial fingerprints? In: 2019 IEEE conference on multimedia information processing and retrieval (MIPR). pp. 506–511. IEEE (2019)
 33. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. In: Proceedings of the

- IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6038–6047 (2023)
34. Navas, K., Ajay, M.C., Lekshmi, M., Archana, T.S., Sasikumar, M.: Dwt-dct-svd based watermarking. In: 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08). pp. 271–274. IEEE (2008)
35. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
36. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning. pp. 8162–8171. PMLR (2021)
37. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv (2022)
38. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 1(2), 3 (2022)
39. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
40. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems* **35**, 36479–36494 (2022)
41. Santana, G.: magicprompt. <https://huggingface.co/Gustavosta/MagicPrompt-Stable-Diffusion> (2022)
42. Sha, Z., Li, Z., Yu, N., Zhang, Y.: De-fake: Detection and attribution of fake images generated by text-to-image diffusion models. arXiv preprint arXiv:2210.06998 (2022)
43. Shaham, T.R., Dekel, T., Michaeli, T.: Singan: Learning a generative model from a single natural image. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4570–4580 (2019)
44. ShieldMnt: invisible-watermark. <https://github.com/ShieldMnt/invisible-watermark/tree/main> (2021)
45. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. arXiv preprint arXiv:2209.14792 (2022)
46. Taha, M.S., Rahem, M.S.M., Hashim, M.M., Khalid, H.N.: High payload image steganography scheme with minimum distortion based on distinction grade value method. *Multimedia Tools and Applications* **81**(18), 25913–25946 (2022)
47. Toderici, G., Shi, W., Timofte, R., Theis, L., Balle, J., Agustsson, E., Johnston, N., Mentzer, F.: Workshop and challenge on learned image compression (clic2020). In: CVPR (2020)
48. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
49. Uchida, Y., Nagai, Y., Sakazawa, S., Satoh, S.: Embedding watermarks into deep neural networks. In: Proceedings of the 2017 ACM on international conference on multimedia retrieval. pp. 269–277 (2017)
50. Vincent, J.: An online propaganda campaign used ai-generated headshots to create fake journalists. *Verge. com* (2020)

51. Vukotić, V., Chappelier, V., Furon, T.: Are deep neural networks good for blind image watermarking? In: 2018 IEEE International Workshop on Information Forensics and Security (WIFS). pp. 1–7. IEEE (2018)
52. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* **13**(4), 600–612 (2004)
53. Wen, Y., Kirchenbauer, J., Geiping, J., Goldstein, T.: Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *arXiv preprint arXiv:2305.20030* (2023)
54. Wu, H., Liu, G., Yao, Y., Zhang, X.: Watermarking neural networks with watermarked images. *IEEE Transactions on Circuits and Systems for Video Technology* **31**(7), 2591–2601 (2020)
55. Yu, N., Davis, L.S., Fritz, M.: Attributing fake images to gans: Learning and analyzing gan fingerprints. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 7556–7566 (2019)
56. Yu, N., Skripniuk, V., Abdelnabi, S., Fritz, M.: Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In: *Proceedings of the IEEE/CVF International conference on computer vision*. pp. 14448–14457 (2021)
57. Yu, N., Skripniuk, V., Chen, D., Davis, L., Fritz, M.: Responsible disclosure of generative models using scalable fingerprinting. In: *International Conference on Learning Representations (ICLR)* (2022)
58. Zhang, C., Benz, P., Karjauv, A., Sun, G., Kweon, I.S.: Udh: Universal deep hiding for steganography, watermarking, and light field messaging. *Advances in Neural Information Processing Systems* **33**, 10223–10234 (2020)
59. Zhang, C., Karjauv, A., Benz, P., Kweon, I.S.: Towards robust deep hiding under non-differentiable distortions for practical blind watermarking. In: *Proceedings of the 29th ACM international conference on multimedia*. pp. 5158–5166 (2021)
60. Zhang, K.A., Xu, L., Cuesta-Infante, A., Veeramachaneni, K.: Robust invisible video watermarking with attention. *arXiv preprint arXiv:1909.01285* (2019)
61. Zhang, L., Liu, X., Martin, A.V., Bearfield, C.X., Brun, Y., Guan, H.: Robust image watermarking using stable diffusion. *arXiv preprint arXiv:2401.04247* (2024)
62. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 3836–3847 (2023)
63. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 586–595 (2018)
64. Zhang, X., Karaman, S., Chang, S.F.: Detecting and simulating artifacts in gan fake images. In: 2019 IEEE international workshop on information forensics and security (WIFS). pp. 1–6. IEEE (2019)
65. Zhao, Y., Pang, T., Du, C., Yang, X., Cheung, N.M., Lin, M.: A recipe for watermarking diffusion models. *arXiv preprint arXiv:2303.10137* (2023)
66. Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: Hidden: Hiding data with deep networks. In: *Proceedings of the European conference on computer vision (ECCV)*. pp. 657–672 (2018)
67. Zohny, H., McMillan, J., King, M.: Ethics of generative ai (2023)