Gaussian Shading: Provable Performance-Lossless Image Watermarking for Diffusion Models

Supplementary Material

1. Details of Gaussian Shading

1.1. Watermark Statistical Test

Detection. Alice embeds a single-bit watermark, represented by k-bit binary watermark $s \in \{0,1\}^k$, into each generated image using Gaussian Shading. This watermark serves as an identifier for her model. Assuming the watermark s' is extracted from image X, the detection test for the watermark can be represented by the number of matching bits between two watermark sequences, $Acc(s, s')$. When the threshold $\tau \in \{0, \ldots, k\}$ is determined, if

$$Acc(s, s') \geq \tau, \tag{1}$$

it is deemed that X contains the watermark.

In previous works [15], it is commonly assumed that the extracted watermark bits $s'_1, \ldots, s'_k$ from the vanilla images are independently and identically distributed, with s'_i following a Bernoulli distribution with parameter 0.5. Thus, $Acc(s, s')$ follows a binomial distribution with parameters $(k, 0.5)$. It is worth noting that if we extract from a vanilla image and decrypt it using a computationally secure stream key [1], the resulting diffused watermark s'^d should be a pseudorandom bit stream, and the corresponding watermark s' would also be pseudorandom. In other words, the bits $s'_1, \ldots, s'_k$ are independently and identically distributed, and each s'_i follows a Bernoulli distribution with a parameter of 0.5. This aligns perfectly with the above assumption.

Once the distribution of $Acc(s, s')$ is determined, the false positive rate (FPR) is defined as the probability that $Acc(s, s')$ of a vanilla image exceeds the threshold τ. This probability can be further expressed using the regularized incomplete beta function $B_x(a; b)$ [5],

$$\begin{aligned} \mathrm{FPR}(\tau) &= \mathbb{P}\left(Acc\left(s, s'\right) > \tau\right) = \frac{1}{2^k} \sum_{i=\tau+1}^{k} \binom{k}{i} \\ &= B_{1/2}(\tau + 1, k - \tau). \end{aligned} \tag{2}$$

Traceability. To enable traceability, Alice needs to assign a watermark $s^i \in \{0,1\}^k$ to each user, where $i = 1, \ldots, N$ and N represents the number of users. During the traceability test, the bit matching count $Acc(s^1, s'), \ldots, Acc(s^N, s')$ needs to be computed for all N watermarks. If none of the N tests exceed the threshold τ, the image is considered not generated by Alice's model. However, if at least one test passes, the image is deemed to be generated by Alice's model, and the index with the maximum matching count is traced back to the corresponding user, i.e., $\mathrm{argmax}_{i=1,\ldots,N} Acc(s^i, s')$. When a threshold τ is given, the FPR can be expressed as follows [5],

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

1.2. Details of Denoising and Inversion

Markov chains of diffusion models. DDPM [7] proposed that the diffusion model consists of two Markov chains used for adding and removing noise. The forward chain is predesigned to transform the data distribution $q_0(x_0)$ into a simple Gaussian distribution $q_T(x_T) \approx \mathcal{N}(x_T|0, \sigma^2 I)$ over a time interval of T. Here, $\sigma > 0$, and the transition probability $q(x_t|x_{t-1})$ is defined as $\mathcal{N}(x_t; \sqrt{\alpha_t}x_0, (1-\alpha_t)I)$, where α_t is a predetermined hyperparameter. By virtue of the Markov property, we have

$$q(x_t|x_0) = \mathcal{N}(x_t|\beta_t x_0, \sigma_t^2 I), \tag{4}$$

with $\beta_t = \sqrt{\overline{\alpha}_t}$, $\sigma_t^2 = 1 - \overline{\alpha}_t$, and $\overline{\alpha}_t = \prod_{i=0}^{t} \alpha_i$.

The transition kernel of the reverse chain is learned by a neural network θ and aims to generate data from a Gaussian distribution with the transition probability distribution defined as

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma(x_t, t)). \tag{5}$$

For LDM [12], since the diffusion process occurs in the latent space $\mathcal{Z}$, Eq. (4) and Eq. (5) should be rewritten for the latent representations z of LDM as follows:

$$q(z_t|z_0) = \mathcal{N}(z_t|\beta_t z_0, \sigma_t^2 I), \tag{6}$$

$$p_\theta(z_{t-1}|z_t) = \mathcal{N}(z_{t-1}; \mu_\theta(z_t, t), \Sigma(z_t, t)). \tag{7}$$

Denoising method for Gaussian Shading. DPM-Solver [10] is a higher-order ODE solver [13], and in this paper, we employ its second-order version during image generation, whose denoising process is as follows,

$$\begin{aligned} v_{t-1} &= t_\lambda\left(\tfrac{\lambda_{t-1}+\lambda_t}{2}\right) \\ u_{t-1} &= \tfrac{\beta_{v_{t-1}}}{\beta_t} z_t^s - \sigma_{v_{t-1}}\left(e^{\frac{h_{t-1}}{2}} - 1\right)\epsilon_\theta\left(z_t^s, c, t\right) \\ z_{t-1}^s &= \tfrac{\beta_{t-1}}{\beta_t} z_t^s - \sigma_{t-1}\left(e^{h_{t-1}} - 1\right)\epsilon_\theta\left(u_{t-1}, c, v_{t-1}\right) \end{aligned}, \tag{8}$$

where $\lambda_t = \lambda(t) = \log\left(\frac{\beta_t}{\sigma_t}\right)$, $t_\lambda(\cdot)$ represents the inverse function of λ_t, $h_{t-1} = \lambda_{t-1} - \lambda_t$, $t = 1, 2, \ldots, T$, and c indicates the prompt used for text-to-image generation.

Noise	Methods					
	DwtDct [3]	DwtDctSvd [3]	RivaGAN [16]	Tree-Ring [14]	Stable Signature [5]	**Ours**
None	0.825/0.881/0.866	**1.000/1.000/1.000**	0.920/0.945/0.963	**1.000/1.000/1.000**	**1.000/1.000/1.000**	**1.000/1.000/1.000**
JPEG	0/0/0	0.013/0.019/0.015	0.156/0.085/0.214	0.997/**1.000**/0.994	0.210/0.217/0.198	**0.999/1.000/0.997**
RandCr	0.982/0.967/0.952	**1.000**/0.998/0.999	0.868/0.878/0.891	0.997/**1.000/1.000**	**1.000**/0.998/0.993	**1.000/1.000/1.000**
RandDr	0/0/0	0/0/0	0.887/0.885/0.862	**1.000/1.000**/0.998	0.971/0.980/0.972	**1.000/1.000/1.000**
GauBlur	0/0.001/0.002	0.430/0.419/0.432	0.328/0.331/0.316	**1.000/1.000**/0.997	0/0/0	**1.000/1.000/1.000**
MedFilter	0/0.001/0.001	0.996/0.999/**1.000**	0.863/0.832/0.873	**1.000/1.000/1.000**	0.001/0/0	**1.000/1.000/1.000**
GauNoise	0.354/0.353/0.364	0.842/0.862/0.884	0.441/0.457/0.535	0/0.006/0.077	0.424/0.406/0.404	**0.996/0.995/0.995**
S&PNoise	0.089/0.160/0.102	0/0/0	0.477/0.411/0.431	0.972/0.986/0.994	0.072/0.078/0.052	**1.000/0.998/0.997**
Resize	0/0.005/0.008	0.985/0.977/0.983	0.850/0.886/0.887	**1.000/1.000/1.000**	0/0/0	**1.000/1.000/1.000**
Brightness	0.126/0.114/0.124	0.110/0.072/0.074	0.480/0.404/0.386	0.084/0.089/0.092	0.843/0.862/0.849	**0.974/0.991/0.979**
Average of Adversarial	0.172/0.178/0.173	0.597/0.594/0.599	0.697/0.697/0.706	0.894/0.898/0.906	0.502/0.505/0.496	**0.997/0.998/0.996**

Table 1. The comparison in the detection scenario. Gaussian Shading demonstrates the best performance.

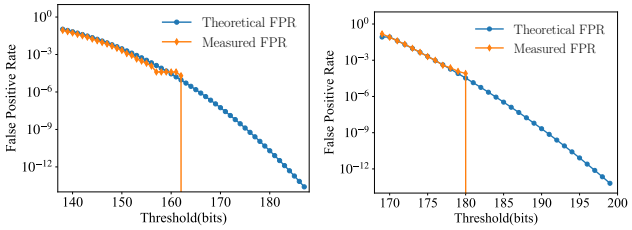
Noise	Methods				
	DwtDct [3]	DwtDctSvd [3]	RivaGAN [16]	Stable Signature [5]	**Ours**
None	0.8030/0.8059/0.8023	0.9997/0.9987/0.9987	0.9762/0.9877/0.9921	0.9987/0.9978/0.9949	**0.9999/0.9999/0.9999**
JPEG	0.5018/0.5047/0.5046	0.5197/0.5216/0.5241	0.7943/0.7835/0.8181	0.7901/0.7839/0.7893	**0.9918/0.9905/0.9872**
RandCr	0.7849/0.7691/0.7673	0.8309/0.7942/0.8151	0.9761/0.9723/0.9735	**0.9933/0.9903/0.9883**	0.9803/0.9747/0.9669
RandDr	0.5540/0.5431/0.5275	0.5814/0.5954/0.6035	0.9678/0.9720/0.9683	**0.9768/0.9747/0.9736**	0.9676/0.9687/0.9649
GauBlur	0.5000/0.5027/0.5039	0.6579/0.6466/0.6459	0.8323/0.8538/0.8368	0.4137/0.4110/0.4112	**0.9874/0.9846/0.9858**
MedFilter	0.5171/0.5243/0.5199	0.9208/0.9287/0.9208	0.9617/0.9585/0.9696	0.6374/0.6399/0.6587	**0.9987/0.9970/0.9990**
GauNoise	0.6502/0.6294/0.6203	0.7960/0.7950/0.8159	0.8404/0.9648/0.8776	0.7831/0.7766/0.7768	**0.9636/0.9556/0.9592**
S&PNoise	0.5784/0.6021/0.5845	0.5120/0.5267/0.5250	0.8881/0.8838/0.8634	0.7192/0.7170/0.7144	**0.9406/0.9433/0.9385**
Resize	0.5067/0.5184/0.5135	0.8743/0.8498/0.8630	0.9602/0.9731/0.9733	0.5278/0.5051/0.5177	**0.9970/0.9975/0.9976**
Brightness	0.5336/0.5097/0.5175	0.5346/0.5234/0.5016	0.8666/0.8496/0.8369	0.9276/0.9267/0.9204	**0.9508/0.9623/0.9527**
Average of Adversarial	0.5696/0.5671/0.5622	0.6920/0.6868/0.6905	0.8986/0.9124/0.9019	0.7520/0.7472/0.7500	**0.9753/0.9749/0.9724**

Table 2. The comparison in the traceability scenario comparison. Although Gaussian Shading slightly underperforms Stable Signature in the presence of Random Crop and Random Drop, considering all the noise, Gaussian Shading still demonstrates the best overall performance.

Inversion method for Gaussian Shading. We note that in DDIM [13], Song et al. proposed an inversion method where they used the Euler method to solve the ODE [13] and obtained an approximate solution for the inverse process:

$$z'^{s}_{t+1} = \sqrt{\alpha_t} z'_t + \left(\sqrt{1-\overline{\alpha}_{t+1}} - \sqrt{\alpha_t - \overline{\alpha}_{t+1}} \right) \epsilon \left(z'_t, c, t \right). \tag{9}$$

According to Eq. (9) it is possible to estimate the noise to be added, which enables latent representation restoration.



(a) Theoretical FPR and measured FPR in detection scenario.

(b) Theoretical FPR and measured FPR in traceability scenario.

Figure 1. Empirical check of the FPR.

2. Experimental Details and Additional Experiments

2.1. Empirical check of the FPR

To test the actual FPR of Gaussian Shading, and to validate the accuracy of Eq. (2) and Eq. (3), we performed watermark extraction on $50,000$ vanilla images from the ImageNet2014 [4] validation set. See Fig. 1, the theoretical and actual measured curves are very close, indicating that the theoretical thresholds derived from Eq. (2) and Eq. (3) can effectively guarantee the actual FPR.

2.2. Details of Comparison Experiments

Watermarking methods settings. To ensure a fair comparison, we set the watermark capacity to 256 bits for DwtDct [3] and DwtDctSvd [3]. As RivaGAN [16] has a maximum capacity of only 32 bits, we retain this setting. The capacity and robustness of Stable Signature [5] are determined by Hidden [19] trained in the first stage. However, in our experiments, we find that Hidden with a capacity of 256 bits did not converge during training. Additionally, if there are too many types of noise in the noise layer, Hidden does not converge either. As an alternative, we use the open-source model of Stable Signature with a capacity of 48 bits[1]. During fine-tuning, we utilize 400 images from the ImageNet2014 [4] validation set, with a batch size of 4 and 100 training steps. Tree-Ring [14] is a single-bit watermark, and we only compare it in the detection scenario. Since its Rand mode is more closely aligned with the concept of performance-lossless, we adopt this setting.

The specific experimental results in both scenarios are shown in Tab. 1 and Tab. 2, respectively. In the detection scenario, the average TPR of Gaussian Shading remains above 0.995 in the presence of noise, surpassing the subpar performance of Tree-Ring by approximately 0.1. In the traceability scenario, the average bit accuracy of Gaussian Shading exceeds 97% against noises, outperforming the second-best method, RivaGAN, by around 7%. In both scenarios, Gaussian Shading exhibits superior performance compared to baseline methods.

The t-test for model performance. To measure the performance bias introduced by the watermark embedding, we apply a t-test to evaluate.

We first generate 50,000/10,000 images using SD V2.1 for each watermarking method, divided into 10 groups of 5,000/1,000 images each. We then calculate the FID [6]/CLIP-Score [11] for each group and compute the average value μ_s. Similarly, we generate 50,000/10,000 watermark-free images using SD V2.1, test the FID/CLIP-Score for 10 groups, and calculate the average value μ_0. For the FID, we randomly select 5000 images from MS-COCO-2017 [8] validation set and calculate the scores using the aforementioned groups. For the CLIP-Score, we utilize OpenCLIP-ViT-G [2] to compute the image-text relevance.

If the model performance is maintained, then μ_s and μ_0 should be statistically close to each other. Therefore, the hypotheses are

$$H_0 : \mu_s = \mu_0, H_1 : \mu_s \neq \mu_0. \tag{10}$$

The statistic t-v is calculated as follows:

$$t\text{-}v = \frac{|\mu_s - \mu_0|}{\sqrt{S^* \cdot (\frac{1}{n_s} + \frac{1}{n_0})}}, \tag{11}$$

[1]The GitHub Repository for Stable Signature

Methods	Metrics	
	FID (t-value ↓)	CLIP-Score (t-value ↓)
Stable Diffusion	25.23±.18	0.3629±.0006
DwtDct [3]	24.97±.19 (3.026)	0.3617±.0007 (3.045)
DwtDctSvd [3]	24.45±.22 (8.253)	0.3609±.0009 (4.452)
RivaGAN [16]	24.24±.16 (12.29)	0.3611±.0009 (4.259)
Tree-Ring [14]	25.43±.13 (2.581)	0.3632±.0006 (0.8278)
Stable Signature [5]	25.45±.18 (2.477)	0.3622±.0027 (0.7066)
Ours	25.20±.22 (**0.3567**)	0.3631±.0005 (**0.6870**)

Table 3. Experimental results of t-test.

where

$$S^* = \frac{1}{n_s + n_0 - 2}\left[(n_s - 1)S_s^2 + (n_0 - 1)S_0^2\right], \tag{12}$$

n_s and n_0 represent the number of testing times, which are both set to 10 in the experiments, and S_s and S_0 represent the standard deviations of the FID/CLIP-Score for watermarked and watermark-free images, respectively.

A lower t-value indicates a higher probability that H_0 holds. If the t-value is larger than a threshold, H_0 is rejected, and model performance is considered to have been affected. The significance level for the test is set to $t\text{-}v_{0.05}(n_s + n_0 - 2) = t\text{-}v_{0.05}(18) \approx 2.101$. In terms of the FID, the t-values of the baseline methods, as depicted in Tab. 3, are all greater than the critical value $t\text{-}v_{0.05}(18) \approx 2.101$, except for Gaussian Shading. Regarding the CLIP-Score, Tree-Ring, Stable Signature, and Gaussian Shading all exhibit competitive results. Note that the CLIP-Score tends to measure the alignment between generated images and prompts, while the FID is solely used to assess image quality. In summary, these baseline methods demonstrate a noticeable impact on the model's performance in a statistically significant manner. On the other hand, Gaussian Shading achieved the smallest t-value, which indirectly confirms its performance-lossless characteristic.

2.3. Details of Ablation Studies

Watermark capacity. The watermark capacity is determined by three parameters: channel diffusion factor f_c, height-width diffusion factor f_{hw}, and embedding rate l. To investigate the impact of these hyperparameters on watermark performance, we first fix l to find an optimal value for f_c and f_{hw}. Experimental results are shown in Tab. 4. Subsequently, we fix f_c and f_{hw} to search for the highest possible l, and the corresponding experimental results are presented in Tab. 5.

Considering all factors, we determine that the optimal solution is $f_c = 1$, $f_{hw} = 8$, and $l = 1$, resulting in a watermark capacity of 256 bits.

Sampling methods. Experimental results about sampling methods under different noises are shown in Tab. 6, and all

Noise	f_c - f_{hw} (k bits)							
	1-2 (4096)	4-1 (4096)	1-4 (1024)	4-2 (1024)	1-8 (256)	4-4 (256)	1-16 (64)	4-8 (64)
None	0.9413	0.9380	0.9985	0.9980	0.9999	0.9999	1.0000	1.0000
JPEG	0.7685	0.7588	0.9204	0.9087	0.9872	0.9866	0.9973	0.9989
RandCr	0.6735	0.6554	0.8177	0.7852	0.9669	0.9457	0.9981	0.9963
RandDr	0.6707	0.6785	0.8239	0.7754	0.9649	0.9444	0.9993	0.9985
GauBlur	0.7217	0.7205	0.8846	0.8832	0.9858	0.9881	0.9996	0.9998
MedFilter	0.8151	0.8104	0.9637	0.9589	0.9990	0.9987	0.9999	1.0000
GauNoise	0.7051	0.6933	0.8502	0.8366	0.9592	0.9539	0.9932	0.9933
S&PNoise	0.6711	0.6661	0.8100	0.7987	0.9385	0.9366	0.9933	0.9914
Resize	0.7904	0.7861	0.9478	0.9438	0.9976	0.9976	0.9999	0.9999
Brightness	0.7558	0.7455	0.8737	0.8619	0.9527	0.9526	0.9829	0.9796
Average of Adversarial	0.7302	0.7238	0.8769	0.8614	0.9724	0.9671	0.9959	0.9953

Table 4. Bit accuracy of Gaussian Shading with different factors f_c and f_{hw}, where $l = 1$.

Noise	l (k bits)			
	2 (512)	3 (768)	4 (1024)	5 (1280)
None	0.9918	0.9502	0.8807	0.8188
JPEG	0.9112	0.8301	0.7635	0.7165
RandCr	0.7766	0.7343	0.6937	0.6586
RandDr	0.8111	0.7545	0.7047	0.6708
GauBlur	0.8730	0.7820	0.7188	0.6783
MedFilter	0.9381	0.8534	0.7823	0.7311
GauNoise	0.8572	0.7854	0.7192	0.6750
S&PNoise	0.8261	0.7478	0.6978	0.6546
Resize	0.9243	0.8397	0.7740	0.7188
Brightness	0.8656	0.8190	0.7480	0.7128
Average of Adversarial	0.8648	0.7940	0.7332	0.6907

Table 5. Bit accuracy of Gaussian Shading with different embedding rates l, where $f_c = 1$ and $f_{hw} = 8$.

Noise	Sampling Methods				
	DDIM [13]	UniPC [18]	PNDM [9]	DEIS [17]	DPMSolver [10]
None	0.9999	1.0000	1.0000	0.9999	0.9999
JPEG	0.9864	0.9797	0.9840	0.9849	0.9872
RandCr	0.9758	0.9395	0.9713	0.9507	0.9669
RandDr	0.9778	0.9642	0.9641	0.9990	0.9649
GauBlur	0.9854	0.9818	0.9886	0.9840	0.9858
MedFilter	0.9990	0.9983	0.9991	0.9991	0.9990
GauNoise	0.9710	0.9264	0.9621	0.9518	0.9592
S&PNoise	0.9302	0.9366	0.9363	0.9424	0.9385
Resize	0.9954	0.9952	0.9980	0.9977	0.9976
Brightness	0.9141	0.9431	0.9452	0.9338	0.9527
Average of Adversarial	0.9706	0.9628	0.9721	0.9715	0.9724

Table 6. Bit accuracy of Gaussian Shading with different sampling methods.

of them exhibit excellent performance with an average bit accuracy of approximately 97% against noises.

2.4. Additional Visual Results

See Fig. 2 and Fig. 3, we present the visual results of different watermarking methods on prompts from the MS-COCO-2017 [8] validation set. From the residual images in Fig. 2, it can be observed that DwtDct [3], DwtDctSvd [3], RivaGAN [16], and Stable Signature [5] introduce noticeable watermark artifacts, leading to a degradation in model performance. As shown in Fig. 3, although Tree-Ring [14] watermark is imperceptible, its embedding may directly impair the image quality. Additionally, it may also introduce changes in the object count and spatial relationships, causing inconsistency with the prompt. In the case of Gaussian Shading, as long as the latent representations where the watermark is mapped remain consistent with that of the original image, no changes occur in the generated image.

To further showcase the visual performance of Gaussian Shading, we present the visual results at multiple embedding rates ranging from 1 to 5 on prompts from Stable-Diffusion-Prompt[2]. See Fig. 4, with the increase in watermark length, the model maintains a good generation quality. Moreover, the diversity and randomness of watermarked images indirectly reflect the performance-lossless characteristic of Gaussian Shading.

[2] Stable-Diffusion-Prompts

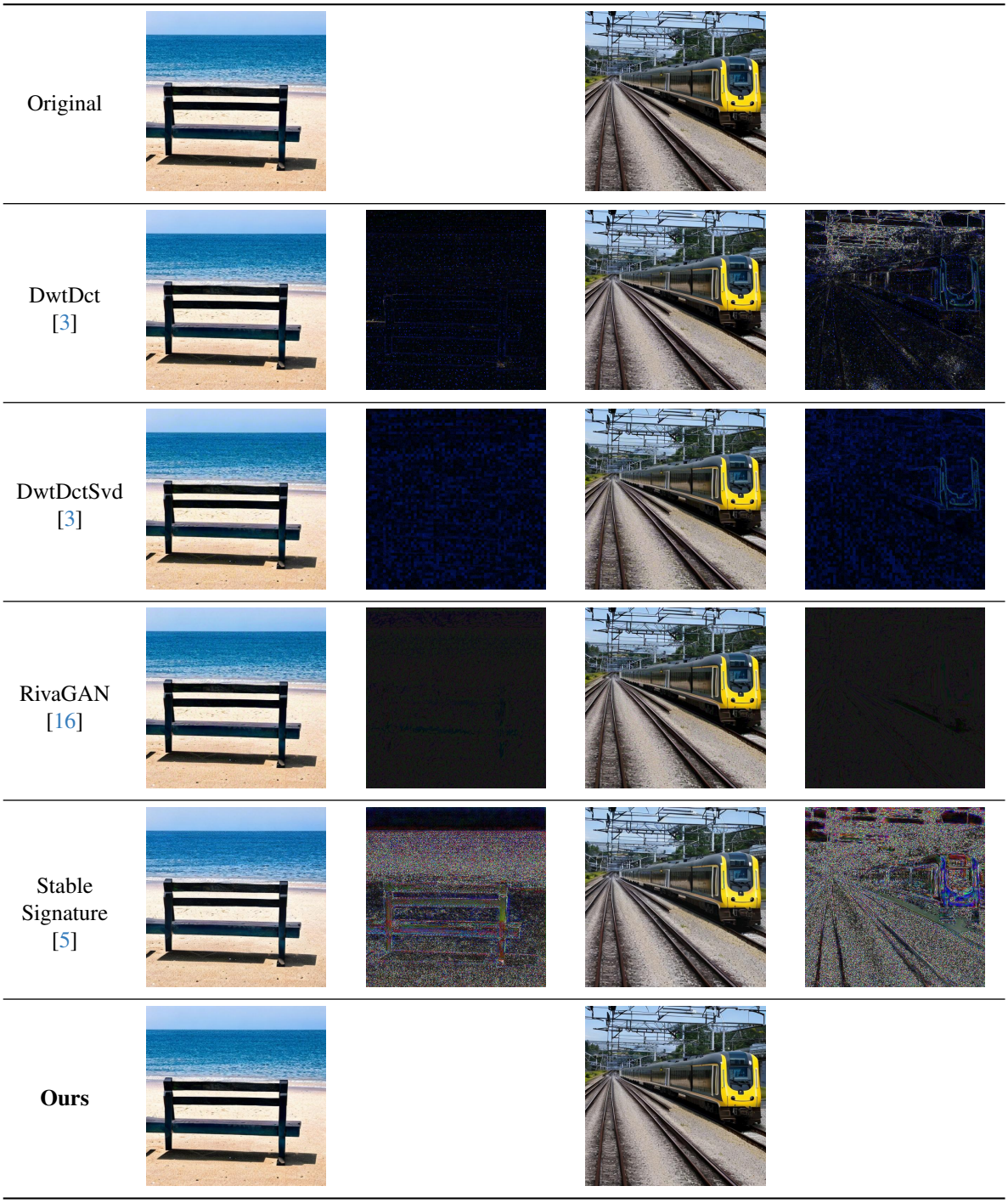


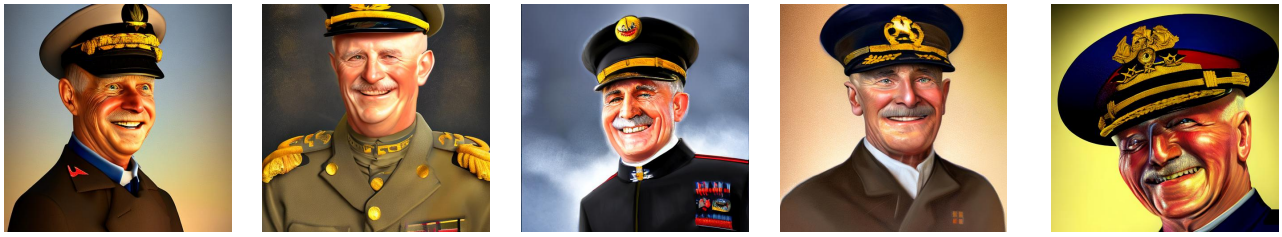
Figure 2. Additional visual results of different watermarking methods, excluding Tree-Ring, on prompts of the validation set of MS-COCO-2017, at resolution 512. All methods are applied with the same input latent representations. Comparison with Tree-Ring is on the next page.

Prompt	Original	Tree-Ring [14]	**Ours**
A bird is sitting on a bowl of birdseed.			
A man holding open an oven door in a kitchen.			
A skillet on a stove with vegetables in it.			
This is two birds pecking at the remnants of a burger at an outdoor restaurant.			
Many surfboards are propped against a rail on the beach.			

Figure 3. Visual comparison between Tree-Ring and Gaussian Shading on prompts of the validation set of MS-COCO-2017, at resolution 512. In contrast to the original model and our Gaussian Shading, Tree-Ring alters the distribution of the latent representations, potentially resulting in the generation of images characterized by semantic inconsistencies or diminished quality. This figure illustrates an instance of such a case, where the Gaussian Shading preserves t.he distribution, thereby avoiding this issue.

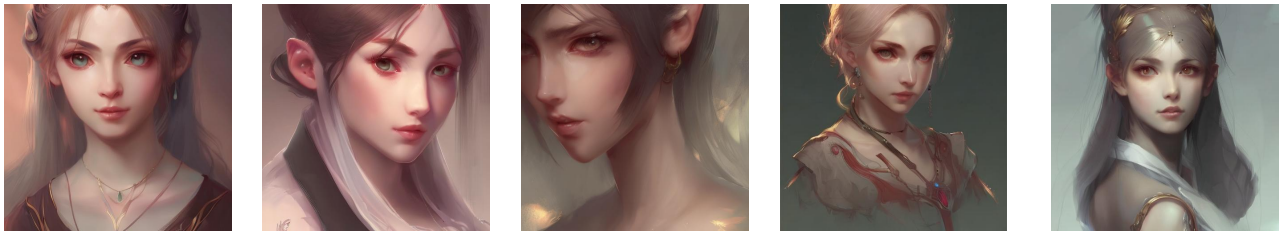
Red dead redemption 2, cinematic view, epic sky, detailed, concept art, low angle, high detail, warm lighting, volumetric, godrays, vivid, beautiful, trending on artstation, by jordan grimmer, huge scene, grass, art greg rutkowski.

Official Portrait of a smiling WWI admiral, male, cheerful, happy, detailed face, 20th century, highly detailed, cinematic lighting, digital art painting by greg rutkowski.



Post apocalyptic city overgrown abandoned city, highly detailed, art by Range Murata, highly detailed, 3d, octane render, bright colors, digital painting, trending on artstation, sharp focus.

A female master, character art portrait, anime key visual, official media, illustrated by wlop, extremely detailed, 8 k, trending on artstation, cinematic lighting, beautiful.



Cat looking at beautiful colorful galaxy, high detail, digital art, beautiful , concept art,fantasy art, 4k.

Figure 4. Additional visual results of Gaussian Shading on generated images at resolution 512. We utilize five prompts in Stable-Diffusion-Prompt and generate images at five different embedding rates l, ranging from left to right as $l = 1, 2, 3, 4, 5$.

References

[1] Daniel J Bernstein et al. Chacha, a variant of salsa20. In *Workshop record of SASC*, pages 3–5. Citeseer, 2008. 1

[2] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023. 3

[3] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital watermarking and steganography*. Morgan kaufmann, 2007. 2, 3, 4, 5

[4] 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. 2, 3

[5] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. *arXiv preprint arXiv:2303.15435*, 2023. 1, 2, 3, 4, 5

[6] 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. 3

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[8] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 3, 4

[9] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022. 4

[10] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 1, 4

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[12] 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

[13] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020. 1, 2, 4

[14] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *arXiv preprint arXiv:2305.20030*, 2023. 2, 3, 4, 6

[15] 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. 1

[16] Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust invisible video watermarking with attention. *arXiv preprint arXiv:1909.01285*, 2019. 2, 3, 4, 5

[17] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022. 4

[18] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *arXiv preprint arXiv:2302.04867*, 2023. 4

[19] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 657–672, 2018. 3