Optimization-Free Universal Watermark Forgery with Regenerative Diffusion Models

Chaoyi Zhu[*1], **ZAITANG LI**[*2], **Renyi Yang**[1],
Robert Birke[4], **Pin-Yu Chen**[5], **Tsung-Yi Ho**[2], **Lydia Y. Chen**[†1,3]
[1]Delft University of Technology, [2]The Chinese University of Hong Kong
[3]University of Neuchâtel, [4]University of Turin, [5]IBM Research
{c.zhu-2, y.chen-10}@tudelft.nl, r.yang-7@student.tudelft.nl
{ztli, tyho}@cse.cuhk.edu.hk, robert.birke@unito.it, pin-yu.chen@ibm.com

Abstract

Watermarking becomes one of the pivotal solutions to trace and verify the origin of synthetic images generated by artificial intelligence models, but it is not free of risks. Recent studies demonstrate the capability to forge watermarks from a target image onto cover images via adversarial optimization without knowledge of the target generative model and watermark schemes. In this paper, we uncover a greater risk of an optimization-free and universal watermark forgery that harnesses existing regenerative diffusion models. Our proposed forgery attack, PnP (Plug-and-Plant), seamlessly extracts and integrates the target watermark via regenerating the image, without needing any additional optimization routine. It allows for universal watermark forgery that works independently of the target image's origin or the watermarking model used. We explore the watermarked latent extracted from the target image and visual-textual context of cover images as prior to guide sampling of the regenerative process. Extensive evaluation on 24 scenarios of model-data-watermark combinations demonstrates that PnP can successfully forge the watermark (up to 100% detectability and user attribution), and maintain the best visual perception. By bypassing model retraining and enabling adaptability to any image, our approach significantly broadens the scope of forgery attacks, presenting a greater challenge to the security of current watermarking techniques for diffusion models and the authority of watermarking schemes in synthetic data generation and governance. Our code is available at the repository: https://github.com/chaoyitud/PnP-Watermark-Forgery.

1 Introduction

The rapid advancement of generative models [1] has led to a significant increase in the creation of high-quality images. As synthetic images proliferate, tracking their provenance [2, 3] is becoming ever crucial. Tracing their usages and verifying their origins are key steps, especially in case of misuse or misconduct. As a result, the demand for effective watermarking techniques to protect intellectual property and ensure the authenticity of generated content has become increasingly important.

Watermarks are a promising solution for embedding information into images, enabling the identification of the creator or source of the content. Post-processing watermarks [4, 5] studied for decades add a covert marker into images as noise on the pixels, but impact visual quality and have low resilience to post-editing attacks. Semantic watermarking [6] overcomes such limits by adding the

[*]Equal Contribution
[†]Corresponding Authors

Preprint. Under review.

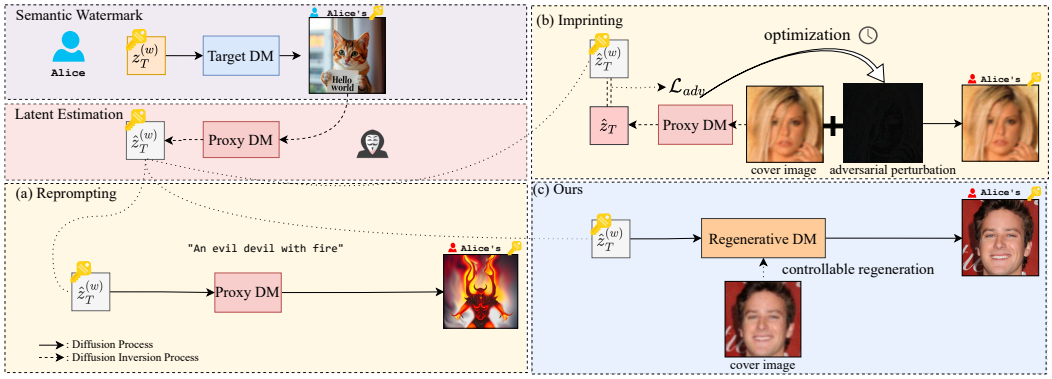


Figure 1: Comparison of different Watermark forgery methods for diffusion models (DM): (a) Reprompt forges the watermarks on text-to-image diffusion models; (b) Imprint forges watermarks on cover images using adversarial optimization on the proxy DM; and (c) PnP, our proposed optimization-free forgery uses controllable regenerative DM.

watermark at the latent space of synthetic data, often during the sampling/generation process. For instance, TreeRing [7] and Gaussian Shading [8] are watermarks for images which are embedded at the initial noise of the diffusion process. This allows the watermark to maintain minimum impact to the image quality, while remaining detectable under various post-editing attacks, e.g., paraphrasing and translation attacks. Furthermore, semantic watermarks can be straightforwardly integrated into existing diffusion models.

Forgery is one of the long standing challenges for data governance, where adversaries can falsify the data without legit authorization. Recent studies [9] demonstrate that the aforementioned semantic watermark for synthetic images can be extracted and forged onto different images by using proxy diffusion models. Reprompt [9], see Fig. 1(a), leverages a proxy diffusion model, specifically text-to-image, and uses different prompts and the extracted "implicit" watermark to generate new images. Imprint [9], see Fig. 1(b), demonstrates a stronger forgery attack on general diffusion models by explicitly embedding the watermark through adversarial optimization. However, reprompting fails to forge watermarks across any images, while imprinting produces watermarked outputs with notable quality degradation and incurs long optimization time, rendering it impractical for large-scale use.

In this work, we unveil Plug-and-Plant (PnP), a watermark forgery attack for images that are generated by diffusion models with either semantic watermarks. Our key proposition is to leverage existing regenerative diffusion models, such as super-resolution [10, 11], face restoration [12], or watermark removal [13], as a backbone for the forgery, without needing any additional finetuning, see Fig. 1(c). PnP allows for universal watermark forgery that works independently of the source image's origin or the watermarking model used. PnP is a light-weight forgery method that uses a controllable regenerative diffusion model to forge the watermark onto either synthetic or real-world images. Specifically, we first extract the watermarked latent from the target image and then explore such latent as prior to regenerate the cover image under its visual-textual guidance as constraints. The forged watermark and key characteristics of cover images are preserved in the generated images. Thus, PnP has two-fold advantages. First, as we are an optimization-free solution, the forged images can be generated up to 50 times faster than existing methods. Secondly, the regenerative diffusion model allows for improved image resolution and quality, overcoming the quality degradation typically associated with adversarial optimization-based methods.

Our findings underline the ease in forging watermarks with off-the-shelf generative models. This allows even users with no forgery expertise to launch such attacks. Hence, model owners cannot reliably depend on current watermarking approaches for data governance, suggesting demands on more advanced and forgery-proof watermarking methods. Our specific technical contributions are:

- We propose, PnP, a plug-and-play universal watermark forgery attack that forges the watermark of a target image onto any arbitrary cover image, by harnessing regenerative diffusion models.

- PnP extracts the forged watermark latents and exploits both the forged latents and visual-textual contexts to condition the regeneration of cover images with the forged watermarks.
- Extensive experimental validation, across multiple state-of-the-art watermarking techniques and regenerative diffusion models, demonstrates that, PnP not only forges watermarks successfully but also improves the overall image perceptuality.

2 Background and Related Studies

2.1 Image Watermarking Methods

Image watermarking methods can be broadly classified into two categories: *post-processing* and *in-processing* methods. Post-processing watermarks modify the final output, embedding the watermark in the image content [4, 5]. In-processing watermarks, on the contrary, are applied during image generation and embed the watermark in the model's internal representation. A unique subcategory of in-processing methods is *semantic watermarking*. In this approach, the watermark is embedded directly into the initial latent space of the image. This latent representation carries the watermark through the image generation process, ensuring its persistence. Moreover, the watermark can be recovered from the generated image using a diffusion inversion process. We focus on two popular semantic watermarks: Tree-Ring [7] and Gaussian Shading [8].

Tree-Ring watermarking is a pioneering semantic watermarking method that embeds a unique pattern into the Fourier domain of the initial latent space. This pattern consists of concentric rings arranged in a tree-like structure. Watermark detection involves the diffusion inversion process, followed by evaluating the distance between the recovered initial latent space and the predefined tree-ring structure in the Fourier domain. An image is watermarked if the distance falls below a model-specific threshold **Gaussian Shading** watermarking embeds a secret code by manipulating the quadrants of the initial latents within the Gaussian distribution. During detection, the watermark is recovered by inverting the diffusion process and checking whether the given value lies within the designated quadrant. Since the watermarked latent follows a Gaussian distribution, the watermarking method achieves near-lossless image quality for a single image. Our forgery attack is effective against both methods.

2.2 Watermark Forgery Attacks

While several related works address watermark forgery attacks, most focus on Tree Ring and do not extend to Gaussian Shading. Currently, only [9] can successfully forge Gaussian Shading watermarks via two novel attacks: *Reprompt* and *Imprint*. Both attacks use DDIM inversion via a proxy diffusion model to obtain the initial latent representation (see Fig. 1). In *Reprompt*, a new image is generated by the diffusion model using the recovered initial latent and a different prompt. This attack is limited to forging watermarks on synthetic images generated by the diffusion model. In *Imprint*, the adversary uses the target image as cover and introduces adversarial noise, which is then optimized through the diffusion process to recover a latent representation similar to that of the watermarked image. Due to the multiple diffusion steps, this optimization process is both computationally intensive and time-consuming. Additionally, the adversarial noise can degrade the quality of the cover image.

Different from previous attacks, we leverage a regenerative diffusion model to embed the watermark into the target cover at low computational cost and superior perceived quality. Moreover, as a side benefit, our method can remove post-processing watermarks from original images before forging the target semantic watermark.

2.3 Regenerative Diffusion Models

Regenerative diffusion models have emerged as a transformative approach for enhancing visual content, with significant advancements in image super resolution and restoration. [11] presents a framework to exploit the diffusion prior for real world image super resolution. It employs a time-aware encoder and feature warping to balance fidelity and perceptual quality by aligning the restoration process with the generative distribution of pre-trained models. [12] introduces DiffBIR, a general restoration pipeline for blind image restoration. By decoupling the process into degradation removal and information regeneration, it enables robust recovery of fine-grained details in degraded images.

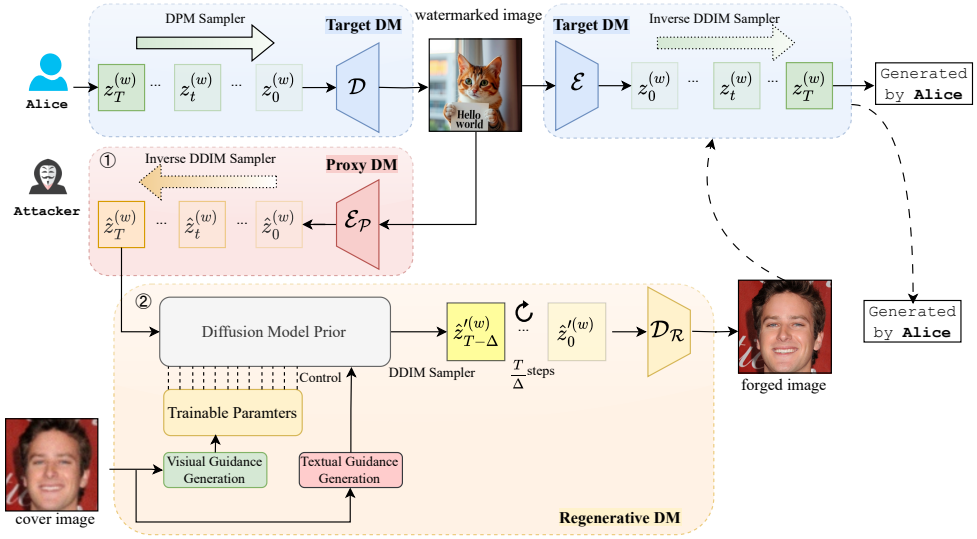


Figure 2: Overview of our watermark forgery approach: PnP. The process consists of two main stages: ① Watermark Latent Estimation and ② Regeneration with Regenerative diffusion models. In the first stage, we estimate the latent representation of the target watermark using a black-box approach based on a publicly available proxy diffusion model. In the second stage, we leverage regenerative diffusion models to manipulate the watermark in a target image without access to the original model parameters.

Semantic-aware super resolution refers to the task of enhancing low-resolution images while integrating high-level semantic information to guide the reconstruction process, ensuring that the upscaled outputs maintain semantic consistency and structural plausibility. For this task, [14] proposes SEESR, which trains a degradation-aware prompt extractor to generate soft and hard semantic prompts; by integrating low-resolution (LR) images into the initial sampling noise, it mitigates the diffusion model's tendency to generate excessive random details, preserving semantic fidelity. [15] focuses on improving the stability of diffusion models for content-consistent super resolution, achieving consistent content across scales through refined latent space manipulation. HoliSDiP [16] advances this field by combining holistic semantics and diffusion prior, using semantic labels as text prompts and segmentation masks for dense guidance to enhance image quality in real world scenarios. These works collectively highlight the versatility of regenerative diffusion models, while also paving the way for addressing remaining challenges in balancing controllability and generative fidelity for more complex real-world applications.

While existing approaches predominantly concentrate on restoration and semantic guidance, SuperMark [17] demonstrates the application of regenerative diffusion models to watermarking tasks. We extend the utility of regenerative diffusion models to watermark forgery, exploiting their inherent latent space manipulation capabilities to achieve seamless embedding of target watermarks into arbitrary images through optimization-free methods.

3 Method - PnP Forgery Attack

In this section, we present PnP, our novel approach for watermark forgery, which leverages regenerative diffusion models to manipulate semantic watermarks without access to the original diffusion model parameters. Our approach, illustrated in Fig. 2, consists of two key stages: ① watermark latent estimation highlighted via the red box; and ② regeneration with regenerative diffusion models enclosed in the orange box. Notice that Stage ① extracts the key characteristics of the watermark ensuring that it is embedded in any image outputted by Stage ②. This stage is only needed once for each watermark to be forged. Once extracted, different cover images provide different new semantic features for the images to be generated by the regenerative diffusion model in Stage ②. In our

example, the cat image contains the watermark of Alice, which we want to embed into the cover profile picture. After regeneration, watermark detection attributes both images to Alice.

Adversarial assumptions. Here, we assume adversaries are curious and have no knowledge about the target model, except being a diffusion model, nor the watermarking method. Adversaries have access to proxy diffusion models and computation capacity to perform inference on the proxy and regeneration diffusion models. No adversarial-optimization or finetuning is required.

3.1 Watermark Latent Estimation

Our approach begins in Stage ① by estimating the latent representation of the target watermark using a black-box approach based on the idea from [9]. Semantic watermarking systems assume security through the secrecy of the diffusion models. However, existing work has shown that watermarking models can be attacked in a black-box setting, where the attacker does not have access to the model parameters or training data.

Denote the target diffusion model as $\Theta = (\mathcal{E}, \mathcal{U}, \mathcal{D})$, where $\mathcal{E}$ is the encoder, $\mathcal{U}$ is the noise predictor, and $\mathcal{D}$ is the decoder. The semantic watermark is typically embedded in the latent noise $z_T^{(w)}$ at the initial time step T, which is then denoised and decoded to produce the final watermarked image $x^{(w)}$.

Take a publicly available diffusion model as the proxy model $\Theta_P = (\mathcal{E}_P, \mathcal{U}_P, \mathcal{D}_P)$. It can be used to forge watermarks without access to Θ. Our approach exploits the shared latent space structure between diffusion models to estimate the watermark's latent representation. Given a watermarked target image $x^{(w)}$ generated by the target model Θ, our goal is to estimate the latent noise $\hat{z}_T^{(w)}$ embedded with the watermark. The procedure is as follows:

1. **Latent Encoding:** The attacker uses the proxy encoder $\mathcal{E}_P$ to map the watermarked image $x^{(w)}$ to the proxy model's latent space:

$$\hat{z}_0^{(w)} = \mathcal{E}_P(x^{(w)}). \tag{1}$$

   Given a watermarked target image $x^{(w)}$ generated by the target model Θ, our goal is to estimate the latent noise $\hat{z}_T^{(w)}$ that encodes the watermark.

2. **Noise Estimation:** The inverse DDIM sampler $\mathcal{I}_{0\to T}$ of the proxy model is applied to estimate the latent noise $\hat{z}_T^{(w)}$ from $\hat{z}_0^{(w)}$:

$$\hat{z}_T^{(w)} = \mathcal{I}_{0\to T}(\hat{z}_0^{(w)}; \mathcal{U}_P). \tag{2}$$

This estimated noise $\hat{z}_T^{(w)}$ captures the semantic information embedded in the watermark, which we leverage in the subsequent stage to forge watermarks.

3.2 Regeneration with Regenerative Diffusion Models

In Stage ② of our watermark forgery approach, as illustrated in Fig. 2, we leverage regenerative diffusion models to manipulate the watermark in a target image without access to the original model parameters. This stage is a crucial part of our framework, consisting of several key components and processes that work together to achieve watermark forgery.

3.2.1 Watermark Latent Estimation Utilization

In the first stage of watermark latent estimation, we utilize this latent representation in the regenerative process. As shown in the figure, the attacker first uses the inverse DDIM sampler of the proxy diffusion model (Θ_P) to estimate the latent noise. This $\hat{z}_T^{(w)}$ captures the semantic information embedded in the watermark, serving as input latents for the regenerative diffusion model.

3.2.2 Regenerative Diffusion Model Architecture and Process

The regenerative diffusion model at this stage integrates a pretrained, publicly available text-to-image (T2I) diffusion model prior $\mathcal{U}_R$, with its parameters kept fixed. To enable watermark regeneration, a

set of lightweight, trainable components $\mathcal{F}_R$ is introduced. The cover image x_c is first processed by an image encoder $\mathcal{E}_R$ to extract visual embeddings $\mathcal{E}_R(x_c)$, which are then passed to the trainable modules. In some variants, large language models (LLMs) or captioning models $\mathcal{T}_R$ are employed to extract textual information $\mathcal{T}_R(x_c)$, providing prompt-based textual guidance to condition the diffusion prior.

During training, the regenerative model aims to reconstruct the cover image at the same or higher resolution by conditioning the generative process on both the visual and textual signals extracted from x_c. The learned modules bridge these signals and guide the generation process, ensuring the preservation of the original image content in a visually consistent manner.

Specifically, the DDIM sampler $\mathcal{G}$ operates over $\frac{T}{\Delta}$ steps, where T denotes the number of sampling steps of the target diffusion model, and Δ is the step length used in the regenerative model. It transforms the estimated watermark latent $\hat{z}_T^{(w)}$ into a regenerated latent representation $\hat{z}_0'^{(w)}$, conditioned on both visual and textual features:

$$\hat{z}_0'^{(w)} = \mathcal{G}_{T\to 0}\left(\hat{z}_T^{(w)} \middle| \mathcal{E}_R(x_c), \mathcal{T}_R(x_c); \mathcal{U}_R, \mathcal{F}_R\right), \tag{3}$$

where $\mathcal{E}(x_c)$ and $\mathcal{L}(x_c)$ provide the visual and textual conditions, respectively. The decoder $\mathcal{D}_R$ then maps the refined latent back to the image space to generate the forged image:

$$x_c^{(w)} = \mathcal{D}_R\left(\hat{z}_0'^{(w)}\right). \tag{4}$$

3.2.3 Transferability and Forgery Realization

The design of the regenerative diffusion model ensures transferability to the target model. By exploiting the shared latent-space structure and diffusion priors across different diffusion models, the regenerative process can effectively manipulate the watermark.

During inference, the combination of visual and textual guidance, along with trainable parameters, steers the denoising trajectory. This adaptation allows the model to regenerate the cover image in a consistent manner, while the preserved diffusion prior and similar denoising process help retain the semantic information of the target watermark.

Specifically, by injecting the estimated watermark latent $\hat{z}_T^{(w)}$ into the regenerative diffusion model and using the associated control mechanisms, such as visual and textual guidance, we can synthesize the appearance of a watermark from the original target model Θ into an arbitrary cover image. This approach operates in a black-box setting, requiring no access to the target model's parameters, and leverages the shared architectural and latent space properties of diffusion models. Our method can be seamlessly integrated into most regenerative diffusion models without retraining or optimization, offering an efficient and generalizable solution for semantic watermark forgery.

4 Evaluation

In this section, we evaluate the watermark forgery risk for synthetic images with watermark signals. We explicitly ask if adversaries can forge the watermark on a target image and replicate to a set of cover images. The criteria for successful forgeries are threefold: (i) detectability of forged watermark and ownership of cover images, (ii) the visual perception of cover images after forgery, and (iii) the overhead of creating forged watermark on the cover images.

4.1 Setup

Datasets To evaluate the performance of our watermark forgery method on various image types, we utilize three datasets: RealSR [18], DRealSR [19], and CelebA [20]. The RealSR and DRealSR datasets consist of real-world photographs, while the CelebA dataset focuses on human face images. All three datasets contain low-quality (LQ) and high-quality (HQ) image pairs, with resolutions of 128×128 and 512×512, respectively.

Table 1: Detectability results of our proposed PnP and baseline on various datasets and target models, evaluated against Gaussian Shading using SD 2.1 as the proxy model. The best results are highlighted in **bold**, and the second-best results are underlined. The watermarking methods are assessed using three metrics: Bit Accuracy (Bit Acc.), Watermark Detection Success Rate (Dec.), and User Attribution Success Rate (Attr.).

Target	Method	Backbone	RealSR			DRealSR			CelebA		
			Bit Acc. ↑	**Dec.** ↑	**Attr.** ↑	**Bit Acc.** ↑	**Dec.** ↑	**Attr.** ↑	**Bit Acc.** ↑	**Dec.** ↑	**Attr.** ↑
SDXL	Imprint	–	**0.93**	**1.00**	**1.00**	**0.94**	**1.00**	**1.00**	0.92	**1.00**	**1.00**
	Ours	CtrlRegen	0.92	**1.00**	**1.00**	0.93	**1.00**	**1.00**	**0.94**	**1.00**	**1.00**
		StableSR	0.85	**1.00**	**1.00**	0.88	**1.00**	**1.00**	0.90	**1.00**	**1.00**
		DiffBIR(SR)	0.84	**1.00**	**1.00**	0.86	**1.00**	**1.00**	0.90	**1.00**	**1.00**
		DiffBIR(Face)	–	–	–	–	–	–	0.87	**1.00**	**1.00**
		SeeSR	0.83	**1.00**	**1.00**	0.86	**1.00**	**1.00**	0.90	**1.00**	**1.00**
		CCSR	0.83	**1.00**	**1.00**	0.84	**1.00**	**1.00**	0.88	**1.00**	**1.00**
		HoliSDiP	0.79	**1.00**	0.99	0.81	0.99	0.99	0.88	**1.00**	**1.00**
PixArt-Σ	Imprint	–	**0.77**	**1.00**	**0.97**	**0.78**	**1.00**	**0.98**	**0.77**	**1.00**	**0.97**
	Ours	CtrlRegen	0.74	0.98	0.88	0.75	0.99	0.88	0.74	**1.00**	0.91
		StableSR	0.65	0.74	0.50	0.69	0.89	0.58	0.69	0.94	0.68
		DiffBIR(SR)	0.67	0.85	0.62	0.68	0.91	0.57	0.71	0.95	0.83
		DiffBIR(Face)	–	–	–	–	–	–	0.68	0.89	0.64
		SeeSR	0.68	0.83	0.60	0.71	0.90	0.73	0.71	0.98	0.79
		CCSR	0.66	0.80	0.53	0.67	0.81	0.56	0.70	0.97	0.78
		HoliSDiP	0.61	0.61	0.24	0.64	0.72	0.39	0.69	0.87	0.62
FLUX.1	Imprint	–	0.76	**1.00**	**0.99**	**0.77**	**1.00**	**0.97**	0.75	**1.00**	0.94
	Ours	CtrlRegen	**0.76**	**1.00**	0.95	0.76	0.99	**0.97**	**0.79**	**1.00**	**0.98**
		StableSR	0.66	0.86	0.62	0.69	0.94	0.77	0.71	0.96	0.89
		DiffBIR(SR)	0.66	0.88	0.59	0.69	0.96	0.75	0.73	0.99	0.96
		DiffBIR(Face)	–	–	–	–	–	–	0.70	0.98	0.81
		SeeSR	0.68	0.98	0.62	0.70	0.96	0.80	0.74	0.98	0.93
		CCSR	0.65	0.81	0.47	0.66	0.81	0.54	0.70	0.97	0.78
		HoliSDiP	0.62	0.69	0.31	0.64	0.72	0.47	0.70	0.92	0.81
Animagine XL	Imprint	–	**0.90**	**1.00**	**1.00**	**0.92**	**1.00**	**1.00**	0.90	**1.00**	**1.00**
	Ours	CtrlRegen	0.90	**1.00**	**1.00**	0.91	**1.00**	**1.00**	**0.91**	**1.00**	**1.00**
		StableSR	0.83	**1.00**	**1.00**	0.85	**1.00**	**1.00**	0.87	**1.00**	**1.00**
		DiffBIR(SR)	0.82	**1.00**	0.99	0.84	**1.00**	0.99	0.86	**1.00**	**1.00**
		DiffBIR(Face)	–	–	–	–	–	–	0.83	**1.00**	**1.00**
		SeeSR	0.80	**1.00**	**1.00**	0.82	**1.00**	**1.00**	0.85	**1.00**	**1.00**
		CCSR	0.80	**1.00**	**1.00**	0.82	**1.00**	**1.00**	0.85	**1.00**	**1.00**
		HoliSDiP	0.75	**1.00**	**1.00**	0.79	**1.00**	**1.00**	0.85	**1.00**	**1.00**

Target and Proxy Models Target models are diffusion models that embed watermarks. We utilize four state of the art diffusion models: Stable Diffusion XL (SDXL) [21], PixArt-Σ [10], FLUX.1 [22], and Animagine XL [23] to generate watermarked images. As our goal is to forge the watermark of the target model on any given image, we employ Stable Diffusion 2.1 [24] as a proxy model to extract the initial latent ($\hat{z}_T^{(w)}$) from watermarked images.

Watermark Schemes We consider two representative semantic watermarking techniques: Tree Ring [7] and Gaussian Shading [8] on the target models.

Regenerative Diffusion Models We consider the following six regenerative models: CtrlRegen [13], StableSR [11], DiffBIR [12], SeeSR [14], CCSR [25], and HoliSDiP [16]. CtrlRegen is a watermark removal method through regenerating images, whereas the other models are designed to increase the image resolution. For the CelebA dataset, we additionally employ the face restoration variant of DiffBIR, fine-tuned on facial datasets and referred to as DiffBIR (Face).

Metrics We consider two types of metrics related to the detectability of forged watermark and the perceptual quality of forged watermark. We define three metrics for detectability: bit accuracy, the detection success rate, and the user attribution success rate for both watermark schemes. Gaussian Shading's bit accuracy is the ratio of correctly detected bits to total bits in the watermark. For TreeRing, the p-value reflects the distance between initial noise and the Tree Ring pattern. The Detection Success Rate (Dec.) is the true positive rate at a given false positive rate: 1e-3 for Gaussian Shading and 1e-2 for TreeRing. For Gaussian Shading , the User Attribution Success Rate (Attr.) is the percentage of the watermark correctly attributed to the user, considering 1000 users.

We utilize a diverse set of image quality metrics to compare between the original image and the image with watermark forgery. PSNR and SSIM [26] are employed as full-reference fidelity measures, assessing pixel-level and structural similarity, respectively. For perceptual quality, we use reference-based metrics LPIPS [27] and DISTS [28]. No-reference metrics CLIPIQA [29], NIQE [30], MUSIQ [31], and MANIQA [32] are adopted to provide a holistic assessment of visual fidelity.

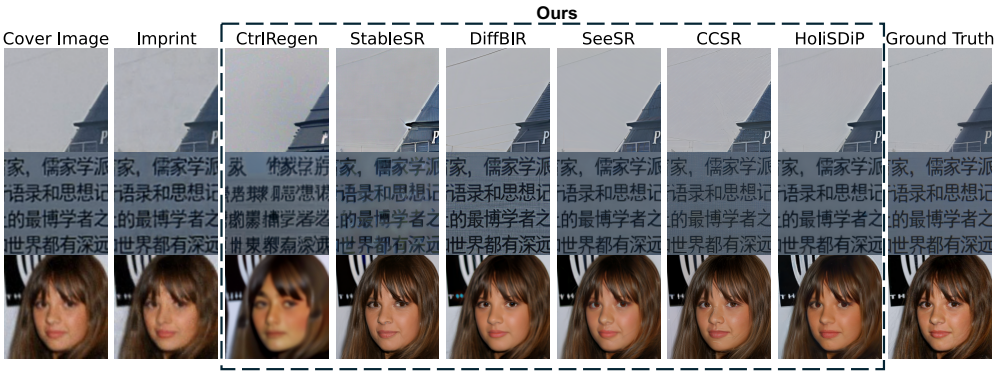
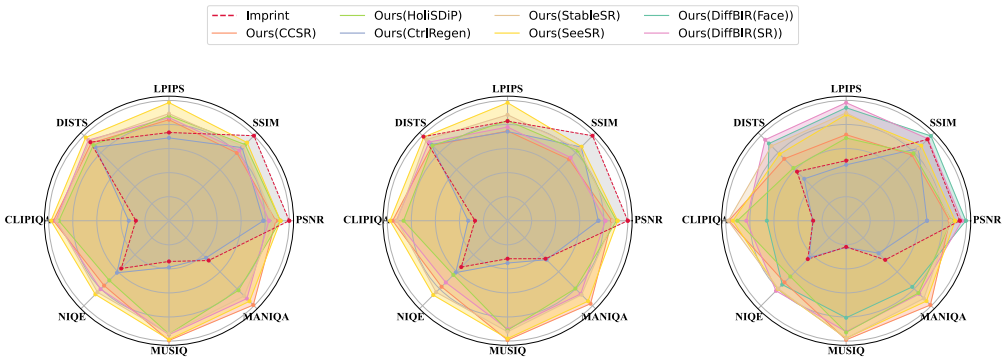


Figure 3: Example images with forged watermark using the Imprint baseline and our proposed method. The target model is SDXL with Gaussian Shading, and the proxy model used is SD 2.1.

4.2 Detecting Forged Watermarks

The success of watermark forgery for Gaussian Shading can be evaluated from three perspectives: the bit accuracy of the forged watermark, its detectability, and the correctness of user attribution. Table 1 summarizes the results against Gaussian Shading across four target models and three datasets. Our proposed attack seamlessly integrates with regenerative methods, and we evaluate it across seven such methods. Overall, the optimization-based Imprint baseline and our PnP with CtrlRegen consistently achieve the best or second-best performance across all three metrics. This can be attributed to the fact that both methods preserve the forged latent watermark without significant alteration. In contrast, the other six regenerative methods impose additional semantic constraints during generation, prioritizing cover image quality. We further analyze the impact of different target models. The proposed PnP performs better on SDXL and Animagine XL than on PixArt-Σ and FLUX.1. This performance gap can be explained by architectural differences: PixArt-Σ and FLUX.1 adopt DiT-based designs, whereas our proxy model, SD2.1, is UNet-based. Furthermore, FLUX.1 introduces a distinct autoencoder architecture, making watermark forgery more challenging. The detectability results for forgery attacks against Tree-Ring can be found in our appendix.



(a) RealSR (b) DRealSR (c) CelebA

Figure 4: Quality of forged watermarks generated by PnP and the Imprint baseline. Metrics are averaged across different target models and watermarks. Scores like LPIPS and NIQE (where lower scores indicate better image quality) are inverted and normalized for consistency.

4.3 Quality of Forged Images

One of the primary shortcomings of state-of-the-art forgery attacks is the degradation of image quality when embedding forged watermarks. In Fig. 3, we visualize cover images with forged watermarks using Imprint and PnP, leveraging different regenerative methods. While Imprint and CtrlRegen demonstrate the best detectability, they suffer from noticeable quality degradation, in contrast to other regenerative models designed to enhance image quality. We hypothesize that the low quality of cover

images with forged watermarks may fail visual inspection or bypass forgery defenses that are yet to be developed.

We summarize eight quantitative metrics for comparing the quality of cover images in Fig. 4. A larger area covered by a forgery attack indicates better image quality and visual perception. For all three datasets, Imprint and CtrlRegen exhibit the smallest covered areas, suggesting poorer quality compared to other super-resolution-based regenerative methods. This aligns with our visual inspection results. Notably, Imprint and CtrlRegen perform worse in non-reference metrics such as CLIPIQA, MUSIQ, NIQE, and MANIQA, further confirming their inferior overall image fidelity. We argue that achieving better performance in non-reference metrics is crucial, as reference cover images may not always be available in practical scenarios.

By integrating the proposed PnP with super-resolution regenerative models, we effectively preserve the semantics of the watermarked latent, along with textual and visual guidance from the cover images. This results in a trade-off, where the detectability of the forged watermark is reduced for improved human imperceptibility, a critical factor for defending against forgery.

4.4 Discussion on Forgery Feasibility

Here, we discuss the feasibility of forgery attacks by measuring the average time required to embed a forged watermark into a single cover image, as illustrated in Fig. 5. Specifically, we evaluate images from the RealSR dataset with SDXL as the target model. Since Imprint relies on adversarial optimization to embed the extracted watermark into each cover image, the average generation time per image on our testbed[3] is approximately 1188.7 seconds. In contrast, the regenerative models used in PnP require only 3.68 to 29.3 seconds per image. This significant discrepancy highlights the prohibitive computational cost of Imprint and questions its practical applicability. Conversely, the low overhead of PnP underscores its effectiveness and practicality as a forgery attack, particularly given that the regenerative models employed are publicly available.

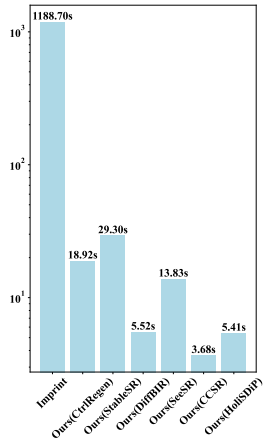


Figure 5: Average overhead required to forge watermark to a cover image.

Our proposed watermark forgery method is both optimization-free and universal, leveraging regenerative diffusion models to perform efficient forgery. The effectiveness of the attack—measured in terms of watermark detectability, visual fidelity, and computational cost—depends heavily on the choice of the underlying generative model. For instance, while StableSR and SeeSR incur higher overhead, methods like DiffBIR, CCSR, and HoliSDiP complete the forgery in roughly 5 seconds. Notably, StableSR, DiffBIR, and SeeSR exhibit superior performance in preserving the forged watermark on the regenerated cover images (see Table 1). In terms of image quality, visual performance varies across datasets, as shown in Fig. 4. Therefore, the optimal regenerative model choice is both data- and model-dependent. The low cost of executing watermark forgeries via PnP further emphasizes the urgency of developing watermarking techniques resilient to such attacks.

5 Conclusion

While watermarks offer an effective mean to manage the governance of synthetic data from generative models, recent studies show that they unfortunately fall prey to forgery attacks under certain restricted circumstances. In this paper, we unveil a stronger watermark forgery attack that leverages off-the-shelf regenerative diffusion models. Our proposed PnP forgery is universally applicable to diffusion models and any type of image. We extract the watermarked latent from the target data and explore both the semantic latent as a prior and textual-visual guidance from cover images to regenerate the cover data. We extensively evaluate the proposed PnP on six regenerative models and 24 combinations of target models, watermarking methods and data sets, against the prior optimization-based Imprint forgery attack. Our results demonstrate that the state-of-the-art watermarks can be successfully forged from a target image to any cover image, which shows a remarkably high detection of forged watermarks

[3]Hardware specifications are provided in Appendix B

and an improved image quality provided by the regenerative models. Our findings raise alarming concerns about the dependability of applying watermarks for data governance and suggest the need for designing more advanced forgery-proof watermarking methods.

Limitations. While `PnP` can be seamlessly integrated into various regenerative diffusion models, it currently requires modifying the initial latent representation to the estimated watermarked latent, and switching the sampling method to DDIM to improve forgery success rates. These adjustments, however, can degrade the overall performance of the regenerative diffusion models. Moreover, achieving different trade-offs between watermark detectability and image quality cannot be easily controlled by simply tuning the hyperparameters of a single regenerative diffusion model.

References

[1] Staphord Bengesi, Hoda El-Sayed, MD Kamruzzaman Sarker, Yao Houkpati, John Irungu, and Timothy Oladunni. Advancements in generative ai: A comprehensive review of gans, gpt, autoencoders, diffusion model, and transformers. *IEEE Access*, 12:69812–69837, 2024.

[2] Yusuf Mehdi. Announcing microsoft copilot, your everyday ai companion. `https://blogs.microsoft.com/blog/2023/09/21/announcing-microsoft-copilot-your-everyday-ai-companion/`, 2023. Accessed: 2025-05-15.

[3] Sven Gowal and Pushmeet Kohli. Identifying ai-generated images with synthid. `https://deepmind.google/discover/blog/identifying-ai-generated-images-with-synthid/`, 2023. Accessed: 2025-05-15.

[4] Shelby Pereira and Thierry Pun. Robust template matching for affine resistant image watermarks. *IEEE Trans. Image Process.*, 9(6):1123–1129, 2000.

[5] Ning Bi, Qiyu Sun, Daren Huang, Zhihua Yang, and Jiwu Huang. Robust image watermarking based on multiband wavelets and empirical mode decomposition. *IEEE Trans. Image Process.*, 16(8):1956–1966, 2007.

[6] Zhengyuan Jiang, Moyang Guo, Yuepeng Hu, and Neil Zhenqiang Gong. Watermark-based attribution of ai-generated content, 2024.

[7] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 58047–58063. Curran Associates, Inc., 2023.

[8] Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12162–12171, 2024.

[9] Andreas Müller, Denis Lukovnikov, Jonas Thietke, Asja Fischer, and Erwin Quiring. Black-box forgery attacks on semantic watermarks for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025.

[10] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023. *URL https://arxiv. org/abs/2310*, 426.

[11] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin C.K. Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. 2024.

[12] Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Bo Dai, Fanghua Yu, Yu Qiao, Wanli Ouyang, and Chao Dong. Diffbir: Toward blind image restoration with generative diffusion prior. In *European Conference on Computer Vision*, pages 430–448. Springer, 2024.

[13] Yepeng Liu, Yiren Song, Hai Ci, Yu Zhang, Haofan Wang, Mike Zheng Shou, and Yuheng Bu. Image watermarks are removable using controllable regeneration from clean noise. *arXiv preprint arXiv:2410.05470*, 2024.

[14] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 25456–25467, 2024.

[15] L Sun, R Wu, Z Zhang, H Yong, and L Zhang. Improving the stability of diffusion models for content consistent super-resolution. arxiv 2023. *arXiv preprint arXiv:2401.00877*.

[16] Li-Yuan Tsao, Hao-Wei Chen, Hao-Wei Chung, Deqing Sun, Chun-Yi Lee, Kelvin CK Chan, and Ming-Hsuan Yang. Holisdip: Image super-resolution via holistic semantics and diffusion prior. *arXiv preprint arXiv:2411.18662*, 2024.

[17] Runyi Hu, Jie Zhang, Yiming Li, Jiwei Li, Qing Guo, Han Qiu, and Tianwei Zhang. Supermark: Robust and training-free image watermarking via diffusion-based super-resolution, 2024.

[18] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3086–3095, 2019.

[19] Pengxu Wei, Ziwei Xie, Hannan Lu, Zongyuan Zhan, Qixiang Ye, Wangmeng Zuo, and Liang Lin. Component divide-and-conquer for real-world image super-resolution. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 101–117. Springer, 2020.

[20] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9168–9178, 2021.

[21] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[22] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[23] Cagliostro Research Lab. Animagine xl 3.0. `https://huggingface.co/cagliostrolab/animagine-xl-3.0`, 2024. Accessed: 2025-05-16.

[24] 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.

[25] Lingchen Sun, Rongyuan Wu, Zhengqiang Zhang, Hongwei Yong, and Lei Zhang. Improving the stability of diffusion models for content consistent super-resolution. *arXiv preprint arXiv:2401.00877*, 2024.

[26] 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.

[27] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[28] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020.

[29] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 2555–2563, 2023.

[30] Lin Zhang, Lei Zhang, and Alan C Bovik. A feature-enriched completely blind image quality evaluator. *IEEE Transactions on Image Processing*, 24(8):2579–2591, 2015.

[31] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021.

[32] Sidi Yang, Tianhe Wu, Shuwei Shi, Shanshan Lao, Yuan Gong, Mingdeng Cao, Jiahao Wang, and Yujiu Yang. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1191–1200, 2022.

A Implementation Details

In this section, we provide implementation details for our target semantic watermarks, target diffusion models, baseline forgery method, regenerative diffusion models, and dataset specifications. Further details can be found in our source code.

A.1 Watermark

For Tree Ring [7], we use a ring pattern with a radius of 10 and apply zero-bit watermarking. For the models used in existing work [9] (SDXL, PixArt-Σ, FLUX.1), we adopt the same detection thresholds established in that study, which were derived from statistics on 5,000 watermarked images and 5,000 clean images to achieve the desired false positive rate. For Animagine XL, we follow the same procedure to compute the detection threshold by ourselves.

For Gaussian Shading [8], we follow the settings from [9]. We use an encoding window of $l = 1$, with a unique random key and message generated for each image. The message length k is 256, resulting in 1024 bits. The repetition factor ρ is set to 64 for SD 2.1 and PixArt-Σ. For FLUX.1, the repetition factor is set to 256, since this model uses 16-channel latents, whereas the previous two models use 4-channel latents. For threshold selection to compute the true positive rate, we target a false positive rate of 10^{-3}. In the Detection Success Rate evaluation (zero-bit scenario), we use a bit accuracy threshold of 0.5976525. For the User Attribution Success Rate evaluation (with 1, 000 users), the bit accuracy threshold is set to 0.6484375.

A.2 Diffusion Pipeline

All models (both *Target* and *Proxy*) are configured with a DDIM scheduler using 50 inference steps and a guidance scale of 7.5. For the watermark forgery experiments, we use the target models (excluding Animagine XL) to generate 100 watermarked images based on the first 100 prompts from the Stable Diffusion Prompts test set[4]. For Animagine XL, since it performs better with structured prompts, we generated the prompts manually. The detailed prompts are available in our repository.

Table 2: Settings of diffusion pipelines used in our experiments.

Model Name	Huggingface ID	Sampler	Steps	Guidance Scale
SD 2.1 [24]	stabilityai/stable-diffusion-2-1-base	DDIM	50	7.5
SDXL [21]	stabilityai/stable-diffusion-xl-base-1.0	DDIM	50	7.5
PixArt-Σ [10]	PixArt-alpha/PixArt-Sigma-XL-2-512-MS	DDIM	50	7.5
FLUX.1 [22]	black-forest-labs/FLUX.1-dev	DDIM	50	7.5
Animagine XL [23]	cagliostrolab/animagine-xl-3.0	DDIM	50	7.5

A.3 Baseline

We compare our method against Imprint [9]. We follow the official implementation and apply 50 steps of adversary adjustment. Parameters of the diffusion pipeline and the watermark setup are kept identical across both methods.

A.4 Regenerative Diffusion Models

We evaluate six regeneration models: CtrlRegen [13], StableSR [11], DiffBIR [12], SeeSR [14], CCSR [25], and HoliSDiP [16]. For each model, we replace the default sampler with DDIM and set the number of sampling steps to 50. All other hyperparameters follow the default configurations provided in their respective official repositories.

A.5 Datasets

We use three datasets as cover images for watermark embedding.

[4] `https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts`

Table 3: Settings of regenerative models used in our experiments.

Model Name	Official Repository	Sampler	Steps
CtrlRegen [13]	`https://github.com/yepengliu/CtrlRegen`	DDIM	50
StableSR [11]	`https://github.com/IceClear/StableSR`	DDIM	50
DiffBIR [12]	`https://github.com/XPixelGroup/DiffBIR`	DDIM	50
SeeSR [14]	`https://github.com/cswry/SeeSR`	DDIM	50
CCSR [25]	`https://github.com/csslc/CCSR`	DDIM	50
HoliSDiP [16]	`https://github.com/liyuantsao/HoliSDiP`	DDIM	50

The **RealSR** [18] test set consists of high-resolution (HR) and low-resolution (LR) image pairs captured by two full-frame DSLR cameras (Canon 5D3 and Nikon D810). It includes 15 pairs for each camera across three scale settings, totaling 90 pairs.

The **DRealSR** [19] test set comprises 93 image pairs, randomly selected at a scale of 4x from five different DSLR cameras.

The **CelebA** [20] test set contains 3,000 CelebA-HQ images from the testing partition. The low-quality (LQ) images are generated by a degradation model using the high-quality images as input. For testing, we randomly select 100 images from this set.

B Hardware Details

All experiments were conducted on a remote server equipped with 8 NVIDIA A800-SXM4-80GB GPUs, 2 Intel Xeon Gold 6326 CPUs (64 cores total), and 128GB of RAM. The system runs Ubuntu 22.04 with Linux kernel 6.2 and uses CUDA 12.2 and driver version 535.86.10. Experiments described in Section 4.4 are run on a single A800 GPU. All methods were evaluated within the same batch under identical system conditions.

C Additional Experimental Results

In this section, we present additional results for detectability against the Tree Ring watermark and detailed quality results for all watermarks and target models.

C.1 Quantitative Results on Detectability Against Tree Ring

Table 4 shows the detectability results for Tree Ring watermarks. It presents the watermark detection success rates for both our method and the baseline across various datasets and target models, evaluated against Tree Ring using SD 2.1 as the proxy model. The Watermark Detection Success Rate is defined as the true positive rate under a 1% false positive rate. From the results, we observe that, for SDXL and Animagine XL, the forged watermark demonstrates better detectability. However, for PixArt-Σ and FLUX.1, the watermarks are almost undetectable. The reason for this is that SDXL and Animagine XL show higher similarity with our target model, SD 2.1. Additionally, the lower robustness of the Tree Ring watermark also contributes to the reduced Watermark Detection Success Rate.

C.2 Quantitative Results on Image Quality after Watermark Forgery

This subsection presents quantitative results for each target model (SDXL, PixArt-Σ, FLUX.1, and Animagine XL), complementing the analysis in Fig. 4. The results are organized by the watermarking method (Tree Ring and Gaussian Shading) and the dataset (realSR, DRealSR, and CelebA), as shown in tables 5 to 10 . We evaluate image quality after watermark forgery using a comprehensive set of metrics. The best results are highlighted in **bold**, and the second-best results are underlined.

Table 4: Detectability results of our method and baseline on various datasets and target models, evaluated against Tree Ring using SD 2.1 as the proxy model. The best results are highlighted in **bold**, and the second-best results are underlined. The watermarking methods are assessed using two metrics: average p-value (p-value), Watermark Detection Success Rate (Dec.).

Target	Method	Backbone	RealSR		DRealSR		CelebA	
			p-value ↓	**Dec.** ↑	**p-value** ↓	**Dec.** ↑	**p-value** ↓	**Dec.** ↑
SDXL	Imprint	–	0.01	0.97	0.01	0.91	0.03	0.76
	Ours	CtrlRegen	**0.00**	**0.98**	**0.00**	**1.00**	**0.00**	**1.00**
		StableSR	0.01	0.92	0.01	0.87	0.02	0.88
		DiffBIR(SR)	0.05	0.73	0.03	0.80	0.04	0.70
		DiffBIR(Face)	–	–	–	–	0.07	0.65
		SeeSR	0.02	0.88	0.02	0.84	0.01	0.86
		CCSR	0.02	0.81	0.01	0.84	0.01	0.90
		HoliSDiP	0.04	0.69	0.02	0.83	0.01	0.94
PixArt-Σ	Imprint	–	0.09	0.51	0.09	0.60	0.13	0.30
	Ours	CtrlRegen	**0.08**	**0.53**	**0.05**	**0.72**	**0.07**	**0.52**
		StableSR	0.21	0.26	0.16	0.37	0.18	0.21
		DiffBIR(SR)	0.16	0.25	0.15	0.37	0.17	0.33
		DiffBIR(Face)	–	–	–	–	0.21	0.20
		SeeSR	0.16	0.37	0.12	0.47	0.11	0.48
		CCSR	0.19	0.32	0.11	0.41	0.12	0.37
		HoliSDiP	0.19	0.23	0.14	0.32	0.13	0.45
FLUX.1	Imprint	–	**0.20**	**0.17**	**0.18**	**0.12**	0.25	0.07
	Ours	CtrlRegen	0.21	0.13	0.21	0.09	0.22	0.06
		StableSR	0.27	0.05	0.22	0.09	0.27	**0.13**
		DiffBIR(SR)	0.29	0.04	0.27	0.06	0.31	0.02
		DiffBIR(Face)	–	–	–	–	0.34	0.07
		SeeSR	0.28	0.09	0.25	**0.12**	0.27	0.04
		CCSR	0.34	0.05	0.35	0.03	0.29	0.04
		HoliSDiP	0.28	0.06	0.28	0.08	**0.19**	0.12
Animagine XL	Imprint	–	0.03	0.79	0.03	0.78	0.06	0.52
	Ours	CtrlRegen	**0.01**	**0.88**	**0.01**	**0.89**	**0.02**	**0.84**
		StableSR	0.04	0.71	0.03	0.76	0.05	0.59
		DiffBIR(SR)	0.09	0.46	0.09	0.44	0.09	0.38
		DiffBIR(Face)	–	–	–	–	0.11	0.30
		SeeSR	0.06	0.64	0.05	0.62	0.05	0.66
		CCSR	0.07	0.53	0.04	0.66	0.04	0.62
		HoliSDiP	0.08	0.51	0.06	0.62	0.04	0.72

D Additional Examples

In this section, we present additional example images generated by our method. Figures 6 and 7 show more images with forged watermarks using different methods, supplementing the examples in Fig. 3. Additionally, Figures 8 and 9 display watermarked images generated from different target models using Gaussian Shading and Tree Ring, respectively.

Table 5: Image quality comparison of PnP and the Imprint baseline on RealSR images with forged Gaussian Shading watermarks.

Target	Method	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	CLIPIQA ↑	NIQE ↓	MUSIQ ↑	MANIQA ↑
SDXL	Imprint	**26.14**	**0.73**	0.48	0.27	0.20	9.13	24.73	0.34
	CtrlRegen	20.45	0.63	0.51	0.28	0.23	8.37	29.07	0.32
	StableSR	22.62	0.64	0.39	0.26	0.67	**5.29**	68.99	0.64
	DiffBIR(SR)	21.53	0.61	0.40	0.26	0.68	5.78	69.75	0.66
	SeeSR	24.13	0.67	**0.35**	**0.25**	0.69	5.80	72.36	0.68
	CCSR	22.99	0.59	0.41	0.27	**0.70**	6.27	**73.00**	**0.72**
	HoliSDiP	23.62	0.64	0.42	0.28	0.69	6.73	71.25	0.62
PixArt-Σ	Imprint	**26.16**	**0.73**	0.48	0.27	0.20	9.09	24.68	0.33
	CtrlRegen	20.68	0.63	0.52	0.30	0.24	8.62	27.57	0.30
	StableSR	23.44	0.66	0.39	0.28	0.68	7.52	67.97	0.62
	DiffBIR(SR)	22.16	0.60	0.42	0.27	0.67	7.26	68.20	0.64
	SeeSR	24.34	0.67	0.37	**0.26**	**0.71**	**6.43**	71.85	0.68
	CCSR	23.48	0.58	0.43	0.28	0.68	7.32	**72.34**	**0.70**
	HoliSDiP	25.37	0.71	**0.37**	0.28	0.61	8.87	63.93	0.53
FLUX.1	Imprint	**26.16**	**0.73**	0.48	0.27	0.20	9.14	24.73	0.33
	CtrlRegen	20.67	0.63	0.51	0.30	0.25	8.19	27.67	0.31
	StableSR	23.00	0.64	0.41	0.28	0.69	7.19	68.61	0.64
	DiffBIR(SR)	21.76	0.60	0.43	0.27	0.68	7.41	68.49	0.66
	SeeSR	24.40	0.68	**0.36**	**0.25**	**0.72**	**6.22**	72.10	0.68
	CCSR	23.58	0.60	0.41	0.27	0.70	7.10	**72.91**	**0.72**
	HoliSDiP	23.56	0.59	0.47	0.30	0.66	8.45	69.33	0.58
Animagine XL	Imprint	**26.14**	**0.73**	0.48	0.27	0.20	9.16	24.72	0.34
	CtrlRegen	20.36	0.62	0.51	0.28	0.23	8.34	28.85	0.32
	StableSR	22.47	0.62	0.39	0.26	0.64	**5.13**	68.90	0.64
	DiffBIR(SR)	21.49	0.61	0.39	0.25	0.68	5.48	70.05	0.66
	SeeSR	24.12	0.67	**0.35**	**0.25**	**0.70**	5.35	72.12	0.68
	CCSR	22.89	0.57	0.42	0.27	0.69	6.31	**72.47**	**0.72**
	HoliSDiP	24.41	0.68	0.38	0.26	0.68	5.95	70.49	0.62

Table 6: Image quality comparison of PnP and the Imprint baseline on DRealSR images with forged Gaussian Shading watermarks.

Target	Method	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	CLIPIQA ↑	NIQE ↓	MUSIQ ↑	MANIQA ↑
SDXL	Imprint	**29.24**	**0.81**	0.47	0.28	0.20	10.53	22.04	0.31
	CtrlRegen	21.91	0.70	0.51	0.29	0.22	9.40	25.13	0.32
	StableSR	25.28	0.68	0.44	0.28	0.68	**6.00**	65.80	0.61
	DiffBIR(SR)	23.39	0.61	0.49	0.29	0.68	7.18	64.72	0.60
	SeeSR	26.65	0.71	**0.39**	**0.27**	**0.72**	6.25	69.38	0.65
	CCSR	25.10	0.58	0.51	0.30	0.71	6.90	**70.14**	**0.68**
	HoliSDiP	25.71	0.65	0.49	0.32	0.67	7.49	67.74	0.59
PixArt-Σ	Imprint	**29.28**	**0.82**	0.47	**0.28**	0.20	10.50	22.23	0.30
	CtrlRegen	22.34	0.71	0.53	0.33	0.24	9.74	23.91	0.30
	StableSR	26.29	0.71	0.44	0.29	0.68	8.50	63.59	0.57
	DiffBIR(SR)	24.36	0.60	0.51	0.31	0.65	9.31	60.94	0.56
	SeeSR	26.74	0.70	**0.41**	0.29	**0.70**	**7.78**	68.09	0.63
	CCSR	25.68	0.59	0.53	0.32	0.67	8.68	**68.43**	**0.64**
	HoliSDiP	28.05	0.75	0.41	0.30	0.55	16.05	57.22	0.46
FLUX.1	Imprint	**29.28**	**0.82**	0.47	0.28	0.20	10.45	22.20	0.30
	CtrlRegen	22.25	0.71	0.53	0.32	0.27	9.09	24.56	0.31
	StableSR	26.00	0.69	0.44	0.29	0.69	7.75	64.81	0.59
	DiffBIR(SR)	24.48	0.61	0.50	0.30	0.67	8.90	61.26	0.58
	SeeSR	27.17	0.72	**0.39**	**0.27**	**0.73**	**6.63**	68.46	0.65
	CCSR	25.79	0.60	0.52	0.30	0.70	7.69	**69.73**	**0.66**
	HoliSDiP	25.66	0.59	0.54	0.33	0.64	9.83	64.83	0.55
Animagine XL	Imprint	**29.26**	**0.81**	0.47	0.28	0.20	10.47	22.17	0.31
	CtrlRegen	21.96	0.70	0.51	0.29	0.22	9.28	25.09	0.32
	StableSR	25.07	0.68	0.44	0.28	0.64	**5.71**	64.85	0.60
	DiffBIR(SR)	23.11	0.61	0.48	0.29	0.69	6.68	65.74	0.61
	SeeSR	26.63	0.70	**0.39**	**0.28**	0.70	5.80	68.95	0.65
	CCSR	25.09	0.58	0.52	0.31	**0.71**	6.51	**70.08**	**0.68**
	HoliSDiP	26.69	0.70	0.44	0.29	0.66	6.73	65.77	0.58

Table 7: Image quality comparison of PnP and the Imprint baseline on CelebA images with forged Gaussian Shading watermarks.

Target	Method	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	CLIPIQA ↑	NIQE ↓	MUSIQ ↑	MANIQA ↑
SDXL	Imprint	26.11	0.73	0.58	0.31	0.22	9.80	16.74	0.35
	CtrlRegen	18.60	0.65	0.63	0.36	0.22	10.04	17.37	0.28
	StableSR	23.86	0.63	0.33	0.19	0.80	**4.80**	76.47	0.69
	DiffBIR(SR)	26.49	0.73	**0.30**	**0.18**	0.67	5.14	71.58	0.64
	SeeSR	25.00	0.67	0.33	0.23	0.77	5.04	76.55	0.71
	CCSR	23.63	0.58	0.40	0.24	**0.81**	5.60	**77.59**	**0.74**
	HoliSDiP	23.37	0.57	0.44	0.30	0.77	5.95	76.73	0.68
	DiffBIR(Face)	**27.36**	**0.75**	0.31	0.20	0.52	5.85	61.84	0.58
PixArt-Σ	Imprint	26.14	0.73	0.58	0.31	0.22	9.84	17.01	0.35
	CtrlRegen	18.66	0.65	0.62	0.38	0.23	10.68	17.06	0.29
	StableSR	24.50	0.66	0.32	0.22	**0.82**	7.19	**76.29**	0.69
	DiffBIR(SR)	26.73	0.73	**0.30**	**0.20**	0.70	**5.81**	72.32	0.65
	SeeSR	25.35	0.68	0.34	0.25	0.77	6.48	75.01	0.70
	CCSR	24.03	0.59	0.42	0.27	0.78	7.20	76.02	**0.73**
	HoliSDiP	26.12	0.68	0.38	0.29	0.69	9.39	64.69	0.56
	DiffBIR(Face)	**27.66**	**0.76**	0.31	0.21	0.56	6.02	64.41	0.59
FLUX.1	Imprint	26.13	0.73	0.58	0.31	0.23	9.81	17.15	0.35
	CtrlRegen	18.64	0.64	0.63	0.38	0.25	10.08	17.31	0.29
	StableSR	24.21	0.65	0.33	0.22	**0.82**	6.76	75.64	0.69
	DiffBIR(SR)	26.76	0.73	**0.30**	**0.20**	0.70	**5.65**	72.28	0.66
	SeeSR	25.22	0.67	0.32	0.23	0.79	5.88	76.27	0.71
	CCSR	24.03	0.59	0.41	0.26	0.80	6.83	**77.03**	**0.75**
	HoliSDiP	23.16	0.52	0.51	0.33	0.74	7.51	72.13	0.65
	DiffBIR(Face)	**27.67**	**0.76**	0.31	0.21	0.56	5.93	63.05	0.59
Animagine XL	Imprint	26.14	0.73	0.58	0.31	0.23	9.79	16.92	0.35
	CtrlRegen	18.49	0.65	0.63	0.35	0.21	10.21	17.43	0.29
	StableSR	23.53	0.63	0.33	0.19	0.78	**4.42**	76.52	0.68
	DiffBIR(SR)	26.54	0.73	**0.29**	**0.18**	0.66	4.97	71.38	0.64
	SeeSR	24.68	0.66	0.34	0.23	0.77	4.93	76.44	0.71
	CCSR	23.63	0.58	0.39	0.24	**0.82**	5.23	**77.97**	**0.75**
	HoliSDiP	24.38	0.63	0.39	0.26	0.78	5.40	75.65	0.67
	DiffBIR(Face)	**27.41**	**0.75**	0.30	0.19	0.53	5.77	62.16	0.59

Table 8: Image quality comparison of PnP and the Imprint baseline on RealSR images with forged Tree Ring watermarks.

Target	Method	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	CLIPIQA ↑	NIQE ↓	MUSIQ ↑	MANIQA ↑
SDXL	Imprint	**26.14**	**0.73**	0.48	0.27	0.20	9.15	24.52	0.34
	CtrlRegen	20.57	0.63	0.50	0.28	0.23	8.24	28.69	0.31
	StableSR	22.62	0.63	0.39	0.26	0.66	5.53	68.54	0.64
	DiffBIR(SR)	21.43	0.60	0.41	0.26	0.68	5.85	69.59	0.66
	SeeSR	24.01	0.66	**0.36**	**0.25**	**0.71**	**5.48**	72.03	0.69
	CCSR	22.79	0.57	0.42	0.27	0.70	6.20	**72.65**	**0.72**
	HoliSDiP	23.95	0.65	0.41	0.28	0.69	6.52	71.05	0.62
PixArt-Σ	Imprint	**26.18**	**0.73**	0.48	0.27	0.20	9.16	24.86	0.33
	CtrlRegen	20.86	0.63	0.51	0.30	0.25	8.53	28.24	0.30
	StableSR	23.47	0.65	0.40	0.27	0.70	7.55	68.19	0.62
	DiffBIR(SR)	21.91	0.59	0.44	0.28	0.68	7.29	68.37	0.65
	SeeSR	24.22	0.66	0.38	**0.27**	**0.71**	**6.49**	71.94	0.67
	CCSR	23.50	0.58	0.43	0.28	0.68	7.50	**71.99**	**0.70**
	HoliSDiP	25.44	0.71	**0.36**	0.27	0.61	8.50	63.20	0.53
FLUX.1	Imprint	**26.15**	**0.73**	0.48	0.27	0.20	9.11	24.76	0.33
	CtrlRegen	20.75	0.64	0.50	0.28	0.26	8.90	28.20	0.30
	StableSR	23.66	0.69	0.35	0.26	0.68	7.51	67.58	0.63
	DiffBIR(SR)	21.93	0.64	0.38	0.26	0.70	6.49	70.30	0.66
	SeeSR	24.74	0.70	**0.33**	**0.24**	**0.70**	**6.01**	71.58	0.67
	CCSR	24.00	0.64	0.38	0.25	0.70	6.21	**72.67**	**0.71**
	HoliSDiP	23.97	0.65	0.42	0.29	0.67	7.63	70.68	0.60
Animagine XL	Imprint	**26.15**	**0.73**	0.48	0.27	0.20	9.17	24.79	0.34
	CtrlRegen	20.39	0.63	0.51	0.28	0.22	8.45	28.13	0.32
	StableSR	22.50	0.63	0.38	0.25	0.62	5.48	67.34	0.63
	DiffBIR(SR)	21.26	0.60	0.40	0.26	0.68	5.67	70.15	0.67
	SeeSR	24.17	0.67	**0.36**	**0.25**	0.69	**5.32**	72.22	0.68
	CCSR	23.01	0.58	0.40	0.27	**0.69**	6.23	**72.43**	**0.72**
	HoliSDiP	24.46	0.69	0.37	0.26	0.68	5.86	69.78	0.62

Table 9: Image quality comparison of PnP and the Imprint baseline on DRealSR images with forged Tree Ring watermarks.

Target	Method	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	CLIPIQA ↑	NIQE ↓	MUSIQ ↑	MANIQA ↑
SDXL	Imprint	**29.25**	**0.81**	0.47	**0.28**	0.20	10.52	22.05	0.31
	CtrlRegen	22.15	0.71	0.52	0.29	0.23	9.40	25.55	0.31
	StableSR	25.38	0.67	0.44	0.28	0.66	5.93	65.69	0.61
	DiffBIR(SR)	23.54	0.61	0.49	0.29	0.70	7.32	64.66	0.60
	SeeSR	26.51	0.70	**0.40**	0.28	0.71	**5.82**	68.76	0.65
	CCSR	24.99	0.57	0.53	0.31	**0.71**	6.93	**70.42**	**0.68**
	HoliSDiP	25.96	0.66	0.48	0.31	0.66	7.32	67.22	0.59
PixArt-Σ	Imprint	**29.33**	**0.82**	0.47	**0.28**	0.19	10.54	22.09	0.30
	CtrlRegen	22.40	0.71	0.54	0.33	0.24	9.64	24.14	0.31
	StableSR	26.26	0.70	0.43	0.29	0.68	8.64	63.89	0.57
	DiffBIR(SR)	24.32	0.59	0.52	0.32	0.65	9.63	60.38	0.57
	SeeSR	26.94	0.71	0.41	0.29	**0.72**	**7.44**	68.56	**0.64**
	CCSR	25.65	0.60	0.52	0.32	0.67	9.00	**69.14**	0.64
	HoliSDiP	28.19	0.75	**0.41**	0.30	0.56	10.98	56.84	0.46
FLUX.1	Imprint	**29.29**	**0.81**	0.46	0.28	0.20	10.47	22.11	0.31
	CtrlRegen	22.43	0.71	0.51	0.30	0.27	9.81	24.72	0.31
	StableSR	27.27	0.76	0.37	0.27	0.68	8.22	61.25	0.57
	DiffBIR(SR)	24.18	0.65	0.45	0.29	0.71	7.77	65.45	0.60
	SeeSR	27.56	0.75	**0.36**	**0.26**	**0.71**	**6.64**	67.76	0.64
	CCSR	26.26	0.64	0.47	0.29	0.71	6.93	**69.85**	**0.67**
	HoliSDiP	26.34	0.67	0.48	0.32	0.67	8.10	66.68	0.57
Animagine XL	Imprint	**29.28**	**0.81**	0.47	0.28	0.20	10.46	22.07	0.31
	CtrlRegen	22.04	0.70	0.52	0.29	0.22	9.33	25.43	0.32
	StableSR	25.14	0.67	0.43	**0.27**	0.62	**5.76**	63.53	0.60
	DiffBIR(SR)	23.20	0.59	0.49	0.29	0.68	6.97	64.57	0.60
	SeeSR	26.25	0.70	**0.39**	0.27	**0.70**	6.08	68.77	0.65
	CCSR	25.08	0.57	0.52	0.31	0.70	6.60	**69.76**	**0.68**
	HoliSDiP	26.82	0.71	0.43	0.29	0.67	6.37	65.99	0.58

Table 10: Image quality comparison of PnP and the Imprint baseline on CelebA images with forged Tree Ring watermarks.

Target	Method	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓	CLIPIQA ↑	NIQE ↓	MUSIQ ↑	MANIQA ↑
SDXL	Imprint	26.11	0.73	0.58	0.31	0.23	9.78	16.74	0.35
	CtrlRegen	18.67	0.65	0.63	0.36	0.22	9.87	16.87	0.28
	StableSR	23.71	0.63	0.33	0.19	0.80	**4.78**	76.63	0.68
	DiffBIR(SR)	26.46	0.73	**0.29**	**0.18**	0.67	4.97	71.51	0.64
	SeeSR	24.65	0.66	0.34	0.23	0.78	5.01	76.78	0.72
	CCSR	23.52	0.58	0.41	0.24	**0.82**	5.64	**77.67**	**0.75**
	HoliSDiP	23.40	0.58	0.44	0.30	0.77	5.85	75.88	0.67
	DiffBIR(Face)	**27.30**	**0.75**	0.31	0.19	0.53	5.80	61.68	0.58
PixArt-Σ	Imprint	26.16	0.73	0.58	0.31	0.23	9.80	17.11	0.35
	CtrlRegen	18.70	0.64	0.62	0.38	0.24	10.50	16.95	0.28
	StableSR	24.48	0.66	0.32	0.22	**0.83**	7.13	76.10	0.69
	DiffBIR(SR)	26.79	0.73	**0.30**	**0.20**	0.70	**5.68**	72.32	0.65
	SeeSR	25.19	0.67	0.34	0.25	0.79	6.72	75.41	0.71
	CCSR	24.01	0.59	0.43	0.27	0.79	7.24	**76.24**	**0.74**
	HoliSDiP	26.26	0.69	0.36	0.28	0.71	9.10	65.45	0.56
	DiffBIR(Face)	**27.66**	**0.76**	0.31	0.20	0.57	5.91	64.33	0.59
FLUX.1	Imprint	26.14	0.73	0.58	0.31	0.23	9.78	16.89	0.35
	CtrlRegen	18.82	0.65	0.61	0.36	0.26	10.68	17.96	0.28
	StableSR	25.25	0.69	**0.29**	0.20	**0.80**	6.82	74.58	0.68
	DiffBIR(SR)	26.84	0.74	0.30	**0.19**	0.65	5.73	70.04	0.64
	SeeSR	25.49	0.70	0.32	0.22	0.75	**5.46**	75.21	0.71
	CCSR	24.56	0.62	0.37	0.23	0.80	6.16	**77.40**	**0.75**
	HoliSDiP	22.95	0.52	0.48	0.32	0.76	6.81	75.41	0.70
	DiffBIR(Face)	**27.65**	**0.76**	0.31	0.20	0.52	6.05	60.65	0.58
Animagine XL	Imprint	26.13	0.72	0.58	0.31	0.22	9.83	16.91	0.35
	CtrlRegen	18.36	0.64	0.63	0.35	0.22	10.21	17.33	0.29
	StableSR	23.23	0.63	0.32	0.19	0.79	**4.33**	76.28	0.68
	DiffBIR(SR)	26.49	0.73	**0.29**	**0.18**	0.66	4.98	71.39	0.64
	SeeSR	24.34	0.65	0.35	0.23	0.77	4.87	76.74	0.71
	CCSR	23.56	0.58	0.39	0.23	**0.82**	5.11	**77.96**	**0.75**
	HoliSDiP	24.60	0.64	0.38	0.26	0.77	5.15	75.65	0.67
	DiffBIR(Face)	**27.37**	**0.75**	0.30	0.19	0.52	5.88	60.84	0.58

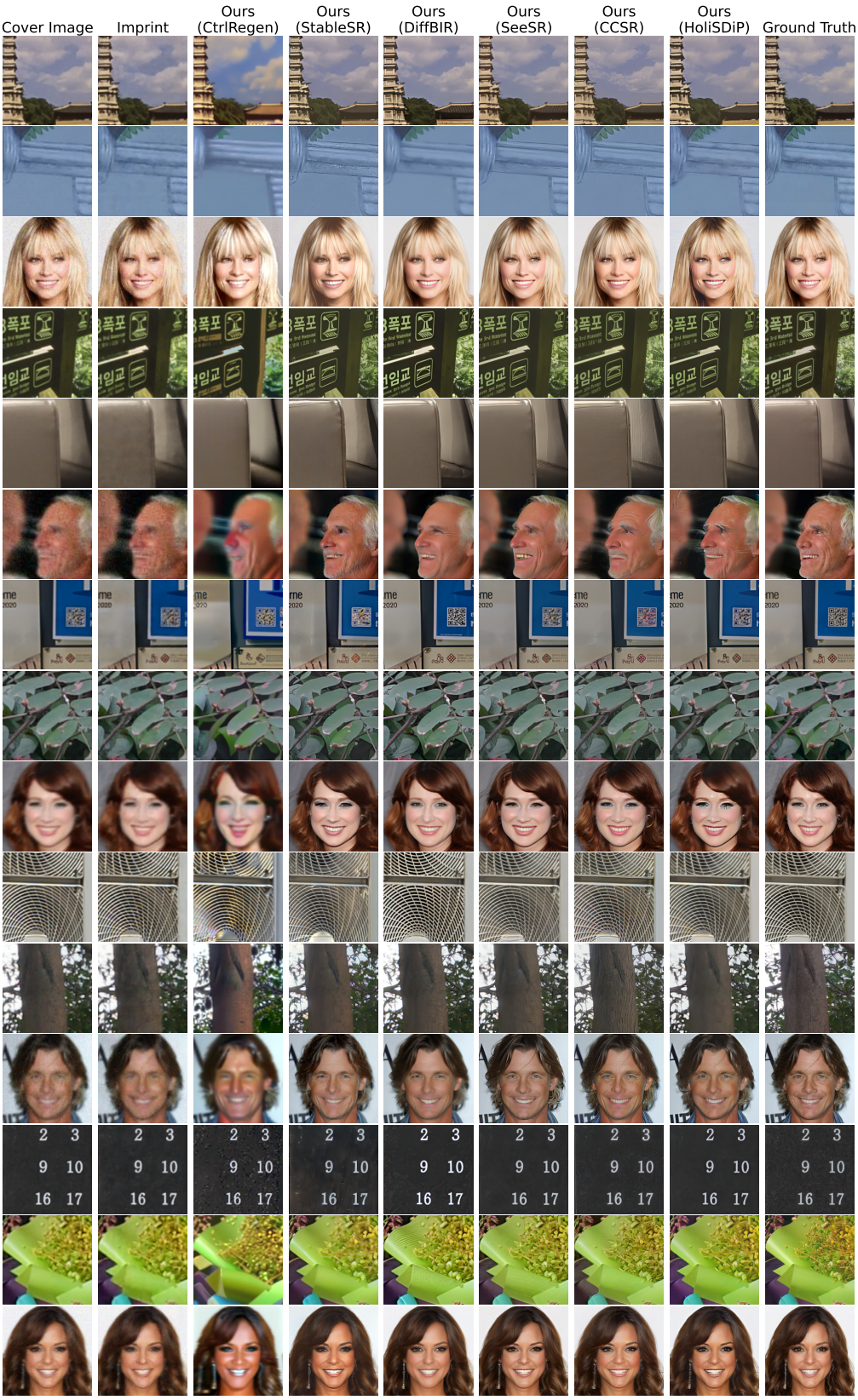


Figure 6: Additional example images with forged watermark using the Imprint baseline and our proposed method.

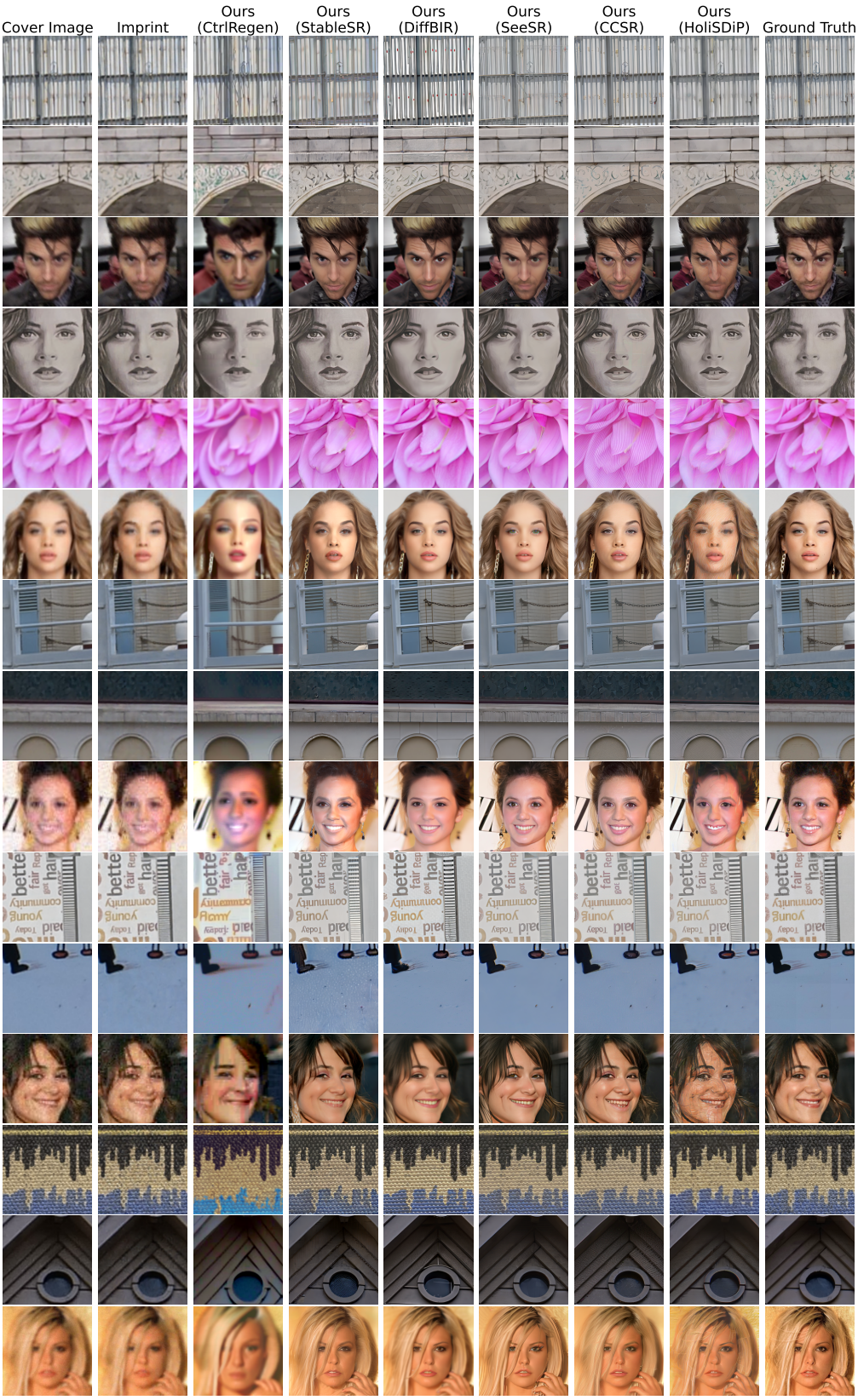


Figure 7: Additional example images with forged watermark using the Imprint baseline and our proposed method.

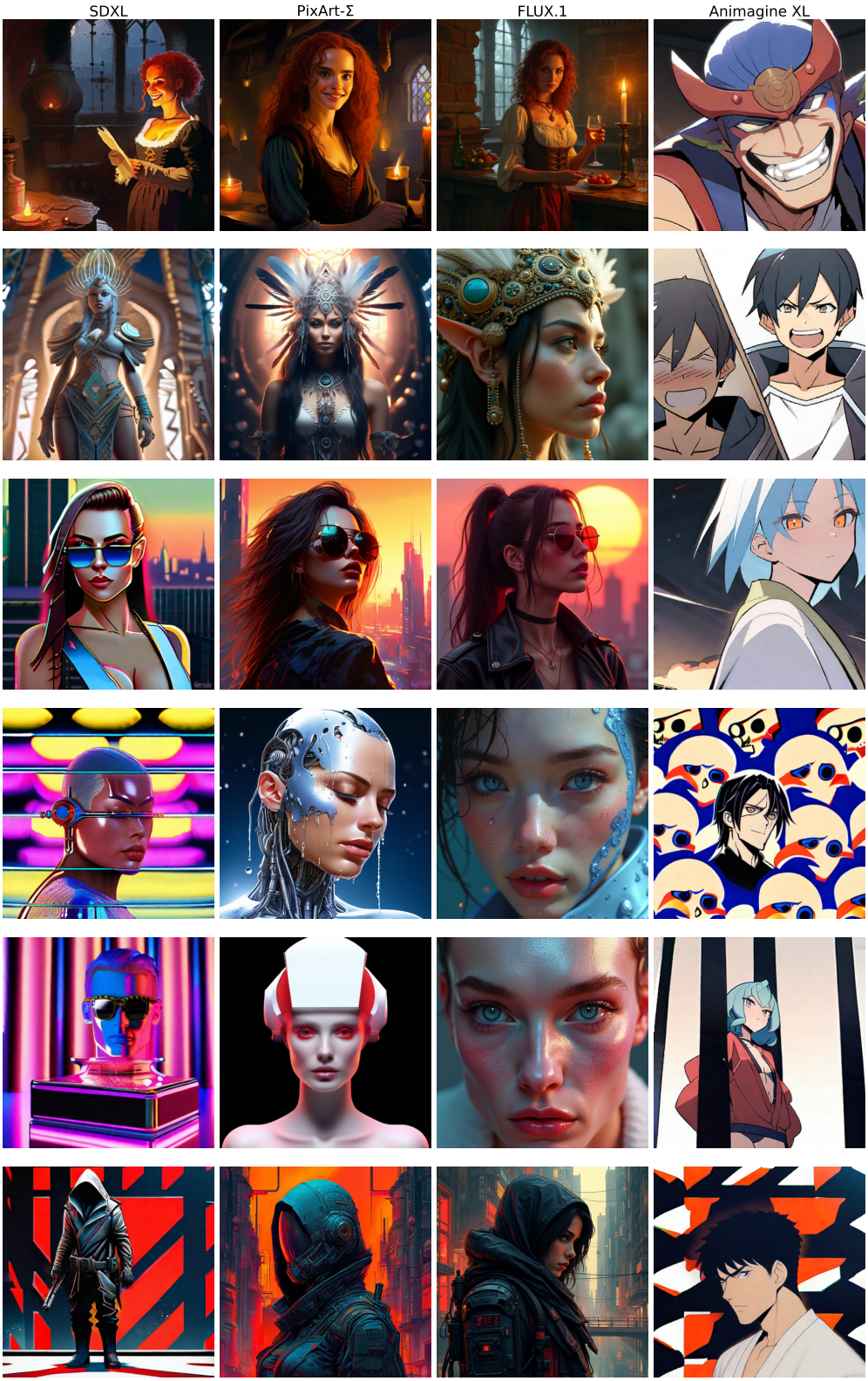


Figure 8: Images generated with the Gaussian Shading watermark using target models.

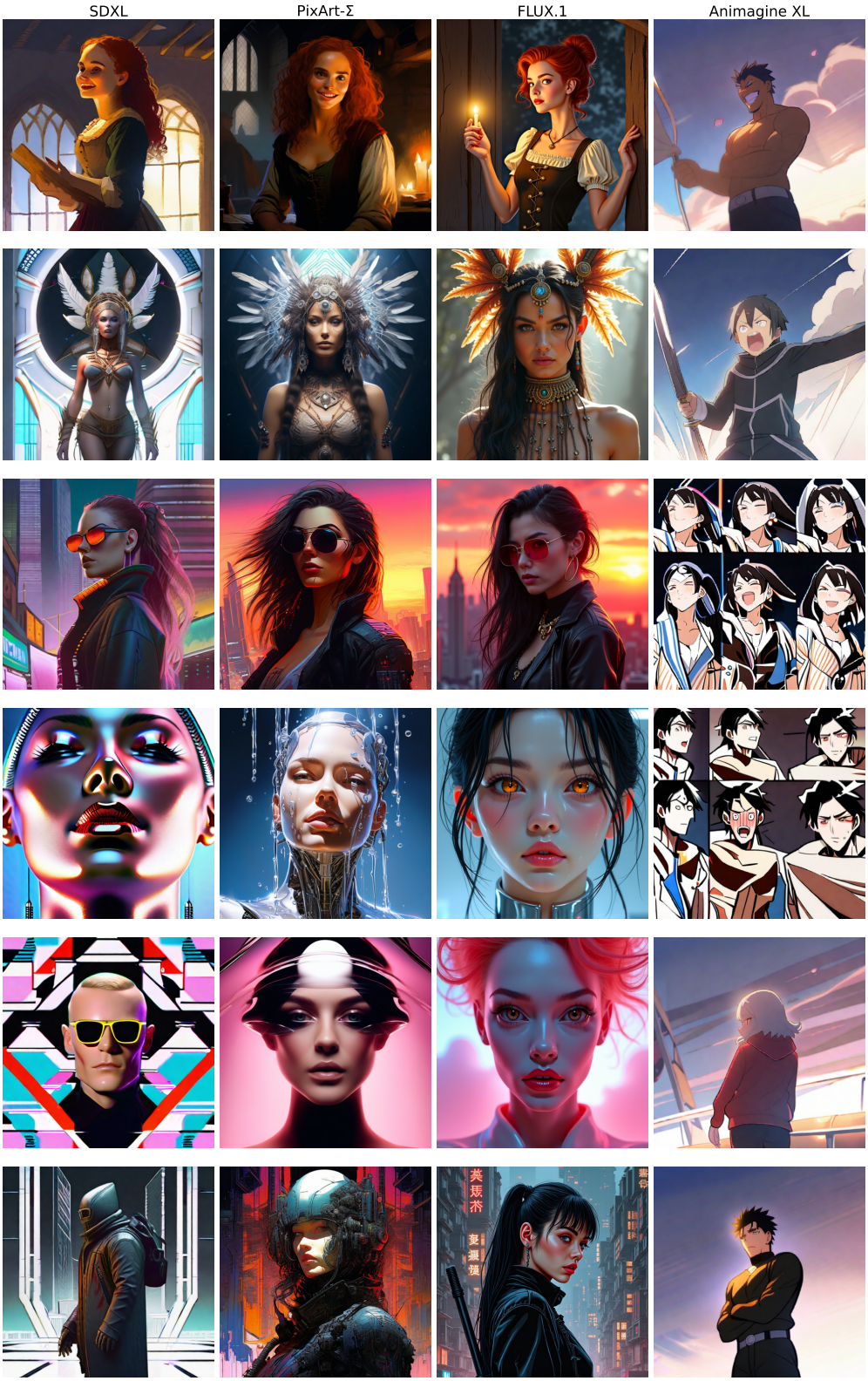


Figure 9: Images generated with the Tree Ring watermark using target models.