Watermarking as a Learned Intrinsic Property of Diffusion Models

Xiang Li[1], **Dong-Dong Wu**[2,3],
Masashi Sugiyama[3,2], **Lannan Luo**[1], **Qiang Zeng**[1,*]
[1]George Mason University
[2]The University of Tokyo
[3]RIKEN Center for Advanced Intelligence Project
[*]Corresponding author

Abstract

Protecting the intellectual property of latent diffusion models is challenging in model distribution scenarios, where downstream users can freely access and modify model weights and generation pipelines. Existing watermarking methods often rely on inference-time control of inputs (e.g., specific prompts or initial noise) or auxiliary watermarking components, making them vulnerable to bypass or removal. We propose INTRIMARK, a data-centric watermarking framework that embeds watermark signals directly into latent-space training data, enabling the denoising network to learn watermarking as an intrinsic property. INTRIMARK remains fully compatible with standard diffusion training pipelines, and the trained network reproduces watermark patterns during generation without relying on inference-time input control or auxiliary watermarking components. Extensive experiments demonstrate that INTRIMARK achieves strong generation fidelity and a true positive rate of 0.995 at a false positive rate of 10^{-6}, while exhibiting robustness against a wide range of image- and model-level attacks. These results highlight the potential of intrinsic watermarking for intellectual property protection in model distribution scenarios. The code is available at `https://github.com/lx913/IntriMark`.

1 Introduction

Recent advances in diffusion models [1, 2], particularly latent diffusion models (LDMs) [3, 4], have enabled the synthesis of highly realistic and diverse visual content [3, 5, 6]. Due to the substantial computational cost and curated data required for training [7, 8, 9], LDMs constitute valuable intellectual property that requires effective ownership protection [10].

Watermarking has emerged as a promising solution for model ownership verification and tracing by embedding identifiable signals into generated outputs [11, 12, 13, 14]. Existing methods explore various strategies, such as modifying the sampling process [13, 15, 16], introducing backdoor triggers [11, 17], or using auxiliary watermarking modules [18, 19, 20]. Despite their differences, these methods often rely on watermark-specific inference inputs (e.g., initial noise or prompts) or dedicated watermarking components during image generation.

However, such dependencies become problematic in model distribution scenarios [14, 21], where downstream users obtain full access to model weights and generation pipelines, allowing arbitrary modifications [14, 21, 17]. In our threat model, the model provider has access only to generated images during ownership verification and therefore cannot enforce specific inference inputs or ensure that auxiliary watermarking components remain active. We illustrate different approaches in Figure 1, and further analyze these methods in detail under model distribution scenarios in Section 2.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

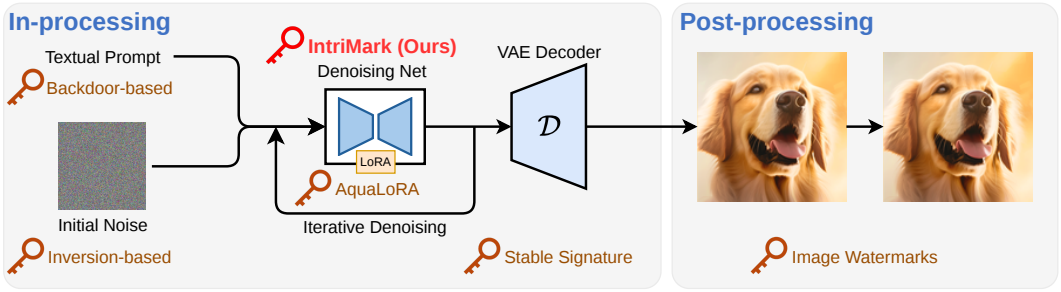

Figure 1: Illustration of different watermarking approaches. Existing methods such as Stable Signature [12] and AquaLoRA [14] rely on specific components (e.g., the VAE decoder or LoRA-based weight modifications), making them susceptible to removal in model distribution scenarios, as demonstrated in Section 6.2. In contrast, INTRIMARK enables the core denoising network to learn watermark patterns through training, making watermarking an intrinsic property of the model.

These limitations reveal a critical gap: existing methods are not well-suited for model distribution scenarios, where robustness to user-side modifications is essential. We address this gap by treating watermarking as an intrinsic property learned by the model, enabling the denoising network to internalize and reproduce watermark patterns without relying on input control or auxiliary components.

However, our first attempt, which adopts a data-poisoning approach by training the model using watermarked images, turns out to be ineffective. Our analysis reveals that directly applying watermarks in pixel space is unsuitable for learning, as watermark signals are largely attenuated during encoding into the latent representation by the VAE encoder. As a result, the denoising network fails to learn to reproduce these signals.

Our insight is that embedding watermark signals in latent space preserves these signals and is well aligned with the training objective of the denoising network. Since the model is trained to predict latent representations, watermark patterns injected at this level can be incorporated into the learned data distribution, enabling reliable watermark reproduction.

Based on this insight, we propose INTRIMARK, a watermarking framework that injects watermark signals into latent-space training data. When trained on these watermarked latents, the denoising network internalizes the watermark as part of the learned data distribution, thereby acquiring an intrinsic watermarking capability. Extensive experiments demonstrate that INTRIMARK achieves strong generation fidelity, high watermark detectability, and robustness against various attacks, while remaining fully compatible with standard diffusion training pipelines. INTRIMARK attains a true positive rate of 0.995 at a false positive rate of 10^{-6}. Even under a range of distortions, such as resizing, noise addition, and compression, it still achieves 0.982 on average, outperforming prior work. Our contributions are summarized as follows:

- We identify a key limitation of existing diffusion watermarking methods in model distribution scenarios: reliance on inference-time input control or auxiliary watermarking components makes them vulnerable to bypass or removal by downstream users.
- We show that watermark signals embedded in pixel-space training images are substantially attenuated during VAE encoding, preventing the denoising network from reliably learning them. We propose INTRIMARK, a data-centric framework that embeds watermark signals directly into latent-space training data, making watermark reproduction an intrinsic capability of the denoising network without modifying standard diffusion training pipelines or relying on inference-time input control or auxiliary watermarking components.
- Our evaluation demonstrates that INTRIMARK achieves strong generation fidelity and a true positive rate of 0.995 at a false positive rate of 10^{-6}, while exhibiting robustness against a wide range of image distortions and practical model-level modifications.

2 Related Work

Existing approaches to watermarking latent diffusion models (LDMs) can be broadly categorized into *post-processing* and *in-processing* methods [14, 18, 21]. Post-processing techniques embed watermarks into generated images after generation [22, 23, 24, 25, 26, 27, 28, 29, 30]. In model distribution scenarios, the watermarking step can be easily removed, for instance by commenting out the corresponding code [14]. In-processing methods incorporate watermarking into the generation process; we further categorize them into *input-control-required* and *input-control-free* methods, and show that existing approaches are insufficient for model distribution scenarios.

Input-control-required Watermarking. *Inversion-based* watermarking methods [13, 15, 16, 31, 32, 33, 34, 35, 21, 36, 37, 38, 39, 40, 41, 42, 43] embed watermark signals into the input sampling noise. For watermark detection, these methods perform inversion [2] to map generated images back to the noise space, where the watermark can be recovered. *Backdoor-based* approaches inject a benign backdoor into LDMs [11, 44, 45, 17, 46, 47, 48]. For instance, prompts containing a special trigger token "*[z]&" produce watermarked images [17]. Another line of work introduces *auxiliary watermarking modules* coupled with LDMs [18, 49, 19, 50, 51, 52, 53, 54, 20, 55, 56, 57], which require explicitly providing a watermark message during generation. Despite their differences, these methods share a common limitation: they rely on injecting watermark-related inputs at inference time. In model distribution scenarios, where downstream users have full control over the generation pipeline, including the input control, such mechanisms can be trivially bypassed.

Input-control-free Watermarking. Stable Signature [12] fine-tunes the VAE decoder to enable watermarking without modifying inputs during generation. AquaLoRA [14] instead couples watermarked LoRA modules [58] with LDMs. However, despite not relying on input control, both methods remain vulnerable to practical removal. For Stable Signature, downstream users can bypass the watermark by replacing the modified VAE decoder with a standard open-source alternative. For AquaLoRA, if the LoRA modules remain unmerged, users can directly remove them with minimal impact on generation quality, as the base denoising network dominates generation quality. Even when the LoRA modules are merged into the model weights, the watermark remains susceptible to removal through LoRA-based fine-tuning. We empirically validate this ease of removal in Section 6.2.

Uniqueness of Our Approach. As shown in Table 1, in contrast to existing approaches, INTRIMARK treats watermarking as an intrinsic property of the diffusion model rather than an external mechanism. Specifically, instead of relying on input control or auxiliary components, INTRIMARK embeds watermark signals directly into the latent training data, enabling the denoising network to internalize and reproduce watermark patterns. This data-centric design fundamentally differs from prior methods that depend on controllable inputs or auxiliary modules, and ensures that watermark generation is enforced by the model parameters rather than the inference pipeline. As a result, INTRIMARK remains effective even when downstream users freely modify inputs or remove auxiliary components, making it inherently robust in model distribution scenarios.

Table 1: Comparison.

Method	Input-control Free?	Robust under model distribution?
Tree-Ring [13]	✗	✗
SleeperMark [17]	✗	✗
Stable Signature [12]	✓	✗
AquaLoRA [14]	✓	✗
INTRIMARK (Ours)	✓	✓

INTRIMARK vs. AquaLoRA. After merging AquaLoRA's watermarked LoRA modules into the denoising model weights, both AquaLoRA and INTRIMARK enforce watermark generation through model parameters. However, INTRIMARK jointly learns image generation and watermark reproduction within the same target latent distribution, whereas AquaLoRA introduces watermarking post hoc through localized low-rank updates after the model has already learned its generation capability. This difference leads to two limitations of AquaLoRA: (1) a trade-off between generation fidelity and watermark extraction accuracy (Appendix K), and (2) vulnerability to downstream LoRA fine-tuning (Section 6.2). In contrast, INTRIMARK avoids AquaLoRA's deployment-time scaling trade-off and exhibits greater robustness to LoRA fine-tuning, distinguishing it from AquaLoRA both methodologically and empirically.

3 Preliminary and Threat Model

Latent Diffusion Models. Latent Diffusion Models (LDMs) learn the data distribution $p(z_0)$, where z_0 is the latent representation of a real image x_0 encoded by a latent encoder $\mathcal{E}$. They are trained by modeling the reverse process of a forward diffusion process defined over a Markov chain of latent variables $\{z_t\}_{t=1}^{T}$. In the forward process, noise is gradually added according to $q(z_t|z_{t-1}) : z_t = \sqrt{1-\beta_t}z_{t-1} + \sqrt{\beta_t}\epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, I)$ and β_t is a predefined variance schedule controlling the noise magnitude at timestep t. With $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, the forward process can be reparameterized as $z_t = \sqrt{\bar{\alpha}_t}z_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. Given such a noisy latent z_t, a denoising model parameterized by θ, typically implemented as a U-Net [59], is trained to predict the added noise ϵ, which implicitly parameterizes the reverse process $p(z_{t-1}|z_t)$. The standard training objective minimizes the mean squared error (MSE) between the predicted noise and the ground-truth noise:

$$\mathcal{L} = \mathbb{E}_{z_0,t,c,\epsilon}\left[\|\epsilon_\theta(z_t, t, c) - \epsilon\|^2\right], \tag{1}$$

where c denotes the conditioning signal, e.g., text embeddings obtained from a pretrained vision-language model such as CLIP [60]. During inference, samples are generated by iteratively applying the learned reverse process starting from Gaussian noise, and the resulting latent z_0 is decoded into pixel space using a latent decoder $\mathcal{D}$.

Threat Model. We consider model distribution scenarios involving two entities: a model provider and downstream users. The model provider embeds a watermark into an LDM for IP protection before distributing the model. Downstream users are granted full access to the model weights and generation pipeline, enabling unrestricted modifications. A malicious user may redistribute, resell, or commercially deploy the model without authorization, and may further claim ownership of the model or deny its provenance. To evade ownership verification, the malicious user can modify the generation pipeline, remove watermarking components, replace model components, or fine-tune the model. During verification, the model provider only has access to images generated by the deployed model, without control over inputs or the generation pipeline, and must determine ownership based solely on these outputs.

4 Method

Motivation and Analysis of a Failed Attempt. Existing watermarking methods are not well-suited for model distribution scenarios, where robustness to user-side modifications is essential. In particular, methods relying on input control or auxiliary watermarking components can be trivially bypassed once downstream users obtain full access to the model weights and generation pipeline. Motivated by this limitation, we explore whether watermarking can instead be learned as an intrinsic property of the diffusion model itself, enabling the denoising network to internalize and reproduce watermark patterns without relying on controllable inputs or auxiliary components.

A natural idea is to directly watermark the entire training dataset, encouraging the diffusion model to generate watermarked images under diverse textual input. We conduct an empirical study following this paradigm. Specifically, we use the COCO2017 dataset [61] and embed watermarks into all training images using a pretrained pixel-space watermark encoder from Stable Signature [12] with a fixed 48-bit watermark signature. We then fine-tune a Stable Diffusion v2.1 model on the resulting watermarked dataset using a standard diffusion training pipeline. During evaluation, images are generated from COCO2017 validation captions and analyzed using the corresponding watermark decoder. However, the resulting bit-wise detection accuracy is only **56.1%**, indicating that watermark signals are not reliably preserved in generated outputs.

To understand this failure, we perform a series of analyses. First, we verify whether the watermark is correctly embedded into the training images. Applying the watermark decoder directly to the watermarked images before diffusion training yields high detection accuracy (**98.51%**), ruling out ineffective embedding as the cause. We then investigate whether watermark signals are degraded during latent encoding. Prior work [14, 62] suggests that pixel-space watermarks may be disrupted when mapped into latent space. To validate this, we encode the watermarked images into latent representations using the latent encoder $\mathcal{E}$ and reconstruct them using the corresponding latent decoder $\mathcal{D}$. Applying the watermark decoder to the reconstructed images reveals a substantial drop in

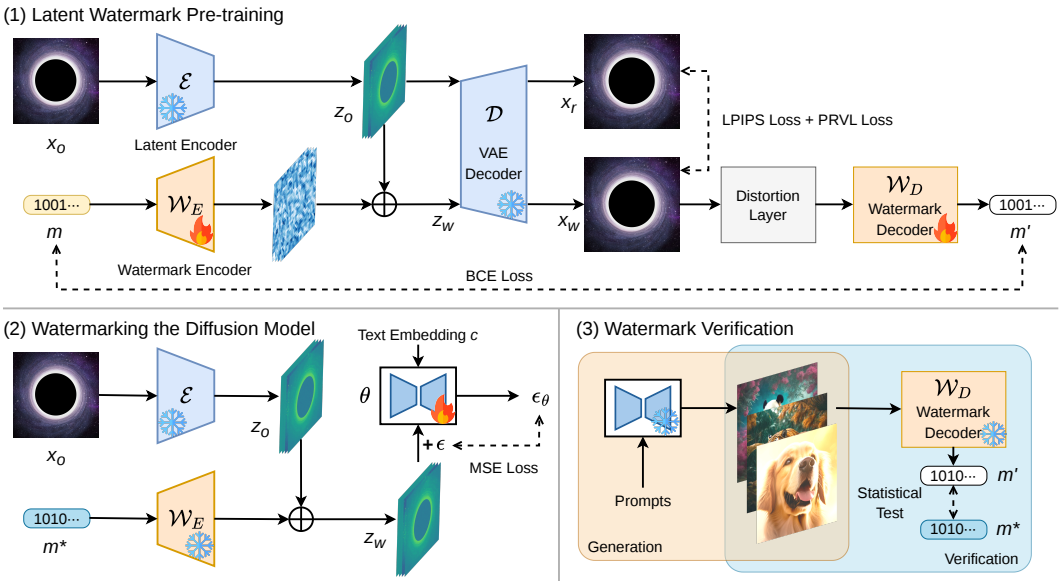

Figure 2: Overview of the INTRIMARK framework.

detection accuracy (**61.46%**), indicating that watermark signals are significantly attenuated during the pixel-to-latent transformation. As a result, the denoising network fails to effectively learn and reproduce the watermark patterns.

Overview of INTRIMARK. Motivated by these observations, we propose INTRIMARK, which embeds watermark signals directly into latent-space training data, enabling the denoising network to internalize and reproduce watermark patterns as part of the learned data distribution. As shown in Figure 2, INTRIMARK consists of three key stages: (1) Latent Watermark Pre-training, where a watermark encoder and decoder are jointly trained in the latent space; (2) Watermarking the Diffusion Model, where watermark signals are embedded into the latent representations of all training images and the diffusion model is trained using the standard objective; and (3) Watermark Verification, where generated images are analyzed using the pretrained watermark decoder.

Latent Watermark Pre-training. In this stage, we jointly train a watermark encoder $\mathcal{W}_E$ and a watermark decoder $\mathcal{W}_D$ for watermark embedding and detection. The encoder $\mathcal{W}_E$ operates in the latent space. Following prior work, we adopt a cover-agnostic watermarking strategy, which improves watermark consistency across samples, and facilitates learning by diffusion models [14, 17, 63]. Specifically, given a k-bit watermark message $m \in \{0,1\}^k$, the encoder $\mathcal{W}_E$ outputs a watermark residual that is added to the latent representation z_o, obtained by encoding the original image x_o with the latent encoder $\mathcal{E}$. The resulting watermarked latent is given by $z_w = z_o + \mathcal{W}_E(m)$. Both z_o and z_w are then decoded into pixel space via the latent decoder $\mathcal{D}$, producing the reference image x_r and the watermarked image x_w, respectively. The watermark decoder $\mathcal{W}_D$ takes watermarked image x_w as input and predicts the recovered message m'. To ensure reliable watermark extraction, we minimize the binary cross-entropy (BCE) loss between m and m'. To preserve visual quality, we additionally apply the LPIPS loss [64] between x_r and x_w. Furthermore, we incorporate the peak regional variation (PRVL) loss [14] to reduce local artifacts and improve fidelity. The overall training objective is defined as

$$\mathcal{L}(\mathcal{W}_E, \mathcal{W}_D) = \mathbb{E}_{x_o,m}\left[\mathcal{L}_{\text{BCE}}(m, m') + \lambda_1 \mathcal{L}_{\text{LPIPS}}(x_r, x_w) + \lambda_2 \mathcal{L}_{\text{PRVL}}(x_r, x_w)\right], \tag{2}$$

where λ_1 and λ_2 balance the contributions of different loss terms.

Watermarking the Diffusion Model. In this stage, we integrate watermark signals into the diffusion model by modifying only the training data, without altering the model architecture or training objective. Specifically, we leverage the pretrained watermark encoder $\mathcal{W}_E$ to inject a fixed watermark message m^* into all training samples. Given an original latent representation z_o, we obtain the corresponding watermarked latent as $z_w = z_o + \mathcal{W}_E(m^*)$. Importantly, the associated text captions

remain unchanged, ensuring that the watermark is learned independently of any specific conditioning input. The diffusion model is then trained on these watermarked latents using the standard diffusion objective. At each timestep t, noise is added to z_w according to the forward diffusion process: $z'_t = \sqrt{\bar{\alpha}_t} z_w + \sqrt{1-\bar{\alpha}_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$. The denoising network ϵ_θ is trained to predict the added noise by minimizing

$$\mathcal{L} = \mathbb{E}_{z_w, t, c, \epsilon}\left[\left\|\epsilon_\theta(z'_t, t, c) - \epsilon\right\|^2\right], \tag{3}$$

where c denotes the conditioning signal (e.g., text prompts). By embedding watermark signals directly into the training latents, the diffusion model is encouraged to internalize the watermark pattern during training, enabling watermark generation without requiring any explicit input control or auxiliary components at inference time.

Watermark Verification. At the verification stage, the model provider only has access to images generated by downstream users, without any control over the inputs or generation pipelines, as defined in our threat model (Section 3). To determine whether an image is generated by the watermarked model, we extract a watermark message m' from the image using the pretrained watermark decoder $\mathcal{W}_D$. Following prior work [12, 14, 17], detection is based on the number of matching bits between the extracted message m' and the target watermark m^*, denoted by $M(m^*, m')$. An image is identified as watermarked if $M(m^*, m') \geq \tau$, where $\tau \in \{0, \ldots, k\}$ is a predefined threshold. This criterion provides robustness to imperfect watermark recovery.

Formally, this decision process can be framed as a hypothesis test between the alternative hypothesis H_1: "*the image is generated by the watermarked model*" and the null hypothesis H_0: "*the image is not generated by the watermarked model*". Under H_0, the extracted bits $m'_1, \ldots, m'_k$ are assumed to be i.i.d. Bernoulli random variables with parameter 0.5, corresponding to random guessing (see Appendix L for empirical validation). As a result, $M(m^*, m')$ follows a binomial distribution with parameters $(k, 0.5)$ [12]. The false positive rate (FPR), defined as the probability of incorrectly identifying a non-watermarked image as watermarked, is given by

$$\text{FPR}(\tau) = \mathbb{P}(M \geq \tau | H_0) = I_{1/2}(\tau, k - \tau + 1), \tag{4}$$

where $I_{1/2}$ denotes the regularized incomplete beta function.

5 Experiments

We present a comprehensive evaluation of INTRIMARK, benchmarking its performance in terms of model fidelity and watermark effectiveness following prior studies [12, 14, 18, 17]. Furthermore, we validate the applicability of INTRIMARK across different datasets, model architectures, and generation paradigms (Appendix C, Appendix D, and Appendix E). A parameter study on sampling scheduler, number of sampling steps, classifier-free guidance, and watermark message length is presented in Appendix F.

5.1 Experimental Setup

Models and Datasets. Following prior work on watermarking latent diffusion models, we implement INTRIMARK on Stable Diffusion v2.1 model (SD v2.1). We train $\mathcal{W}_E$ [14] and $\mathcal{W}_D$ [65] on 10,000 images randomly sampled from the COCO2017 training dataset [61]. For watermarking the diffusion model, we fine-tune SD v2.1 on the full COCO2017 training split (approximately 118k image-caption pairs), following prior studies [20, 19]. (Experiments on other datasets and models are presented in Appendix C and Appendix D.)

Implementation Details. The watermark message length is set to 48 bits (i.e., $k = 48$), consistent with prior studies [12, 14, 17]. During the latent watermark pre-training stage, $\mathcal{W}_E$ and $\mathcal{W}_D$ are jointly trained using AdamW optimizer with a learning rate of 1×10^{-3}, weight decay 1×10^{-4} for 40 epochs. The loss weights λ_1 and λ_2 are set to 10 and 0.5, respectively. We incorporate a distortion layer during training to improve robustness to normal image distortions (see Appendix A.1 for details). During watermarking the diffusion model stage, we fine-tune SD v2.1 using AdamW with a learning rate of 1×10^{-4} for 50 epochs and adopt the standard DDPM scheduler [1].

Table 2: Comparison between INTRIMARK and baseline methods in terms of model fidelity, watermark effectiveness, and robustness. Detailed robustness results are provided in Table 3.

Method	Fidelity		Effectiveness		Robustness	
	FID ↓	CLIP ↑	Bit Acc.↑	T@10^{-6}F↑	Bit Acc. (Adv.)↑	T@10^{-6}F (Adv.)↑
None	26.87	0.314	–	–	–	–
Stable Signature	24.98	0.314	**99.59**	**0.998**	92.49	0.867
AquaLoRA	26.45	0.144	97.17	0.994	95.22	0.977
INTRIMARK	**24.42**	**0.320**	97.39	0.995	**95.78**	**0.982**

Generation and Detection. Following prior work, we evaluate INTRIMARK on text-to-image generation using 5,000 captions from COCO2017 validation set. By default, we adopt DPM-Solver Multistep (DPM-M) [66] scheduler with 30 sampling steps and a classifier-free guidance scale of 7.5.

Comparison Methods. We compare INTRIMARK with Stable Signature [12] and AquaLoRA [14], which are representative input-control-free watermarking methods for diffusion models (Section 2). Like our method, these methods do not rely on control over inputs during inference, making them suitable for comparison. We reproduce both methods based on their official implementations.

5.2 Watermark Performance

This section presents results regarding model fidelity and watermark effectiveness. Robustness results are presented in Section 6.

Fidelity. We adopt FID [67] and CLIP Score [68] to assess the impact on generative capability. FID measures the distribution discrepancy between generated and real images, while CLIP Score evaluates the semantic alignment between text prompts and generated images. We compute FID between generated images and the COCO2017 validation set using `pytorch-fid`, and compute CLIP Score using the ViT-B/32 model from OpenAI's CLIP framework [68].

As shown in Table 2, INTRIMARK achieves the best overall fidelity among all methods, obtaining both the lowest FID and the highest CLIP Score. Compared to Stable Signature and AquaLoRA, our method not only preserves but slightly improves generative quality. We attribute this advantage to the fact that INTRIMARK strictly follows the standard diffusion training pipeline, without introducing architectural modifications or auxiliary modules. By embedding watermark signals directly into the training data, the model naturally incorporates the watermark into the data distribution, thereby maintaining high fidelity. We show examples of generated images in Appendix B.

Effectiveness. We evaluate watermark effectiveness by extracting the embedded message from generated images using the watermark decoder. Following prior work, we report two metrics: bit accuracy (Bit Acc.) and true positive rate at a false positive rate of 10^{-6} (T@10^{-6}F). Bit accuracy is computed as the average matching-bit ratio over all 5,000 generated images. As shown in Table 2, INTRIMARK achieves competitive watermark effectiveness compared to existing methods (e.g., T@10^{-6}F=0.995), demonstrating reliable watermark embedding and extraction.

More Evaluations. We further validate the applicability of INTRIMARK across different datasets, model architectures, and generation paradigms. Despite being trained on COCO, INTRIMARK can still generate watermarked images under captions from out-of-distribution prompts (Appendix C). In addition, the data-centric design allows INTRIMARK to be naturally extended to pixel-space diffusion models (Appendix D). Moreover, INTRIMARK maintains strong performance under the image-to-image generation setting, further highlighting its versatility (Appendix E).

6 Robustness

In this section, we evaluate the robustness of INTRIMARK against various watermark removal attacks.

Table 3: Comparison between INTRIMARK and baseline methods under different image distortions. Overall, INTRIMARK demonstrates the best performance, as shown in the "Average" column.

Method	Distortions							
	ColorJitter	Crop	Resize	Blur	Noise	JPEG	Sharpness	Average
Bit Acc. (%) ↑								
Stable Signature	**98.24**	**99.34**	93.52	93.84	75.21	88.42	**98.85**	92.49
AquaLoRA	94.97	93.07	96.78	96.78	93.16	95.54	96.23	95.22
Ours	95.49	93.49	**97.37**	**97.37**	**93.73**	**96.26**	96.74	**95.78**
T@10^{-6}F ↑								
Stable Signature	**0.995**	**0.997**	0.968	0.957	0.277	0.877	**0.998**	0.867
AquaLoRA	0.972	0.952	0.992	0.993	0.957	0.986	0.986	0.977
Ours	0.976	0.963	**0.995**	**0.994**	**0.966**	**0.991**	0.992	**0.982**

6.1 Image-level Attacks

Image Distortions. We evaluate watermark robustness under a range of image distortions. Specifically, we apply widely used transformations [14, 18, 17], including color jitter, cropping, resizing, Gaussian blur, Gaussian noise, JPEG compression, and sharpness, to 5,000 generated images, and measure both bit accuracy and T@10^{-6}F. Detailed distortion settings are provided in Appendix A.2.

We report the average results in Table 2 (Bit Acc. (Adv.) and T@10^{-6}F (Adv.)) and detailed results for each distortion in Table 3. As shown in the results, INTRIMARK achieves the best overall robustness across different distortions. In particular, it demonstrates strong resilience under challenging distortions such as Gaussian blur, Gaussian noise, and JPEG compression. For other distortions, it also shows stable and consistent watermark recoverability across diverse perturbations.

VAE-based Regeneration. Image compression and regeneration via pretrained VAEs are commonly used operations that can degrade watermark signals [69, 70, 71]. Following SERUM [18], we adopt the model from [69] to perform VAE-based regeneration with the quality level set to 1. Specifically, we apply compression and reconstruction to 5,000 generated images, and then perform watermark extraction on the reconstructed images. We report the resulting bit accuracy and T@10^{-2}F in Table 4, under the "Regen-VAE" setting. Both INTRIMARK and AquaLoRA [14] demonstrate strong robustness against VAE-based regeneration, maintaining high watermark detectability. In contrast, Stable Signature almost completely fails under this attack.

Diffusion-based Regeneration. Recent work has shown that applying a pretrained diffusion model to perform noise addition and denoising on watermarked images can effectively remove watermark patterns [62]. To evaluate robustness against such attacks, we follow the WAVES benchmark [72] and consider two representative strategies: *Regen-Diff* and *Rinse-Diff*. In Regen-Diff, watermarked images undergo multi-step noise addition followed by denoising, while in Rinse-Diff, this process is repeated multiple times to further suppress watermark signals.

We implement these attacks using a pretrained Stable Diffusion v1.5 model with a DDIM scheduler and a 50-step diffusion schedule. Specifically, we evaluate four configurations. Regen-Diff-5 and Regen-Diff-10 apply 5-step and 10-step regeneration, respectively; Rinse-Diff-4×5 applies a 5-step regeneration process and repeats it four times in total, while Rinse-Diff-2×10 applies a 10-step regeneration process and repeats it twice. The results, reported in Table 4, show that INTRIMARK and AquaLoRA maintain a certain degree of robustness against diffusion-based regeneration, whereas Stable Signature fails under all configurations.

6.2 Model-level Attacks

Ease of Watermark Removal. Although Stable Signature and AquaLoRA are also input-control-free, their practical robustness is limited by the ease of watermark removal under model-level attacks. For Stable Signature, the watermark is embedded in the modified VAE decoder, which can be bypassed by simply replacing it with a standard open-source decoder. We validate this by substituting the decoder with the widely used `sd-vae-ft-mse` released by Stability AI [73], after which the watermark signal nearly disappears (Bit Acc. = 46.74%, T@10^{-6}F = 0).

Table 4: Robustness of INTRIMARK and baseline methods against image-level regeneration attacks. Overall, INTRIMARK demonstrates the highest robustness, as reflected in the "Average" column.

Method	Regenerations					
	Regen-VAE	Regen-Diff-5	Regen-Diff-10	Rinse-Diff-4×5	Rinse-Diff-2×10	Average
Bit Acc. (%) ↑						
Stable Signature	50.30	49.33	48.50	47.30	47.44	48.57
AquaLoRA	86.94	**89.13**	**79.67**	66.27	64.64	77.33
INTRIMARK	**87.35**	88.98	78.80	**68.06**	**66.24**	**77.89**
T@10^{-2}F ↑						
Stable Signature	0.007	0.005	0.004	0.001	0.002	0.004
AquaLoRA	0.995	**0.997**	**0.959**	0.543	0.457	0.790
INTRIMARK	**0.999**	**0.997**	0.954	**0.622**	**0.542**	**0.823**

For AquaLoRA, the watermark is embedded through additional LoRA modules, and its robustness depends on how these modules are deployed. When the watermarked LoRA modules remain unmerged, they can be directly removed without modifying the base denoising network. Since the base denoising network dominates generation quality, removing these modules has negligible impact on visual quality. Our experiments confirm this: removing the LoRA modules leads to a complete collapse of watermark effectiveness (Bit Acc. = 46.42%, T@10^{-6}F = 0), while maintaining similar FID (25.55).

Table 5: Robustness of AquaLoRA and INTRIMARK to LoRA-based fine-tuning.

	Step 10	Step 20	Step 30
AquaLoRA	69.88%	51.23%	49.37%
INTRIMARK	97.32%	97.17%	97.36%

Although AquaLoRA allows its watermarked LoRA modules to be merged into the denoising model weights to prevent direct removal, its post-hoc low-rank parameter updates remain vulnerable to subsequent LoRA-based fine-tuning, which is more resource-efficient than full-model fine-tuning and thus represents a practical watermark-removal setting. Under LoRA fine-tuning with rank $r = 16$ on COCO, AquaLoRA's Bit Acc. rapidly decreases from 69.88% at 10 steps to 51.23% at 20 steps and 49.37% at 30 steps. In contrast, INTRIMARK, which jointly learns image generation and watermark reproduction through data-centric training, maintains consistently high Bit Acc. of 97.32%, 97.17%, and 97.36% under the same settings, respectively. These results demonstrate INTRIMARK's greater robustness to subsequent LoRA-based fine-tuning, highlighting the advantage of integrating watermark signals through data-centric training over post-hoc low-rank parameter updates.

Latent VAE Replacement. Downstream users may attempt to remove watermarks by substituting the VAE decoder with alternative implementations, as multiple open-source VAE decoders are readily available. Following prior studies [14, 17], we evaluate this attack using three widely adopted alternatives: the improved decoder `sd-vae-ft-mse` released by StabilityAI [73], the community fine-tuned `ClearVAE` [74], and the `ConsistencyDecoderVAE` introduced by OpenAI [75]. To simulate this attack, we regenerate 5,000 images using each alternative VAE decoder. Watermark extraction is then performed on the resulting images using the same fixed watermark decoder, without knowledge of which VAE decoder was used for generation. The results are reported in Table 6. As shown, INTRIMARK maintains consistently high watermark detection performance. This indicates that the watermark signals learned by INTRIMARK are not tied to a specific decoder implementation, and cannot be effectively removed by simply substituting the VAE component.

Table 6: Impact of VAE replacement on watermark effectiveness and model fidelity.

VAE	Bit Acc.↑	FID↓
`sd-vae-ft-mse`	97.35	24.40
`ClearVAE`	97.34	26.22
`Consistency`	96.95	24.03

Full-model Fine-tuning Attack. Although INTRIMARK demonstrates robustness to LoRA-based fine-tuning, downstream users with access to the model weights may still attempt to remove the watermark through full-model fine-tuning. In model distribution scenarios, however, customized LDMs are often trained on proprietary datasets that are inaccessible to downstream users. In practice, users typically rely on public datasets for fine-tuning, which may differ significantly from the original training distribution. As a result, even if fine-tuning weakens the watermark, it is likely to introduce degradation in generative quality.

To evaluate the impact of training data distribution on watermark removal, we fine-tune the watermarked model using three datasets: COCO, the original dataset used to customize the watermarked model; CC3M [76], a large-scale natural image-text dataset with a distribution relatively close to COCO; and Naruto [77], a dataset with a substantially different distribution from COCO, following SleeperMark [17]. At each fine-tuning checkpoint, we generate images using COCO validation prompts and measure watermark bit accuracy and FID against COCO validation images.

Table 7: Changes of watermark effectiveness (Bit Acc. (%); higher is better) and model fidelity (FID; lower is better) under fine-tuning across datasets.

Dataset	Metric	Step 5	Step 10	Step 15	Step 20
COCO	FID	24.45	25.38	24.33	24.55
	Bit Acc.	78.19	60.43	56.98	53.07
CC3M	FID	24.51	25.82	26.23	26.88
	Bit Acc.	81.41	63.35	56.70	54.15
Naruto	FID	24.55	33.32	37.00	38.45
	Bit Acc.	79.73	65.40	58.42	55.16

As shown in Table 7, fine-tuning on Naruto substantially reduces watermark bit accuracy from 79.73% to 55.16%, while increasing FID from 24.55 to 38.45. A similar but less pronounced trade-off is observed with CC3M, where bit accuracy decreases from 81.41% to 54.15%, accompanied by a smaller FID increase from 24.51 to 26.88. In contrast, fine-tuning on the original COCO dataset reduces bit accuracy from 78.19% to 53.07%, while FID remains nearly unchanged (24.45 to 24.55). These results demonstrate that fine-tuning can weaken the watermark, but using data that differ from the original training distribution introduces degradation in generation quality. While such degradation can be largely avoided by fine-tuning on the original training data, these data may be proprietary or otherwise inaccessible to downstream users in practical model distribution scenarios. Furthermore, fine-tuning large-scale diffusion models requires substantial computational resources [7, 8, 9], which may further limit the practicality of such attacks.

More Potential Attacks. We further evaluate the robustness of INTRIMARK against additional model-level attacks, including model collusion (Appendix G), quantization (Appendix H), and distillation (Appendix I). The results show that INTRIMARK remains robust under these scenarios, demonstrating resistance to a range of practical model-level manipulations.

7 Discussion

Limitation. Although INTRIMARK is effective for model tracing in model distribution scenarios, its support for multi-user tracing remains limited. Specifically, distinguishing among downstream users requires assigning a unique watermark message to each user, which would require training a separate model for each watermark identity. This incurs substantial training overhead and limits scalability as the number of users increases. To mitigate this limitation, we provide a practical provider-side watermark overwriting strategy that enables more efficient assignment of distinct watermark identities to downstream users without training separate models from scratch, as detailed in Appendix J.

Conclusion. In this paper, we identify that existing watermarking methods for latent diffusion models are not robust in model distribution scenarios, where downstream users can freely modify the generation pipeline or remove auxiliary watermarking components. To address this limitation, we propose INTRIMARK, a data-centric watermarking framework that embeds watermark signals directly into latent-space training data, enabling the denoising network to learn watermarking as an intrinsic property. Extensive experiments demonstrate that INTRIMARK achieves strong generation fidelity, reliable watermark detectability, and robustness against various distortions and removal attacks. Overall, our results highlight the potential of intrinsic watermarking for practical model ownership verification in diffusion model distribution scenarios.

Acknowledgments and Disclosure of Funding

This work was supported in part by NSF under Grant CNS-2309550, as well as the Commonwealth Cyber Initiative (CCI), an investment in the advancement of cyber R&D, innovation, and workforce development. Masashi Sugiyama was supported by JST ASPIRE Grant Number JPMJAP25B1. This work also utilized computational resources provided by the Office of Research Computing at George Mason University, supported in part by the NSF under Award No. 2018631.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Annual Conference on Neural Information Processing Systems*, 2020.

[2] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[3] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[4] 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. In *International Conference on Learning Representations*, 2024.

[5] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, 2022.

[6] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Annual Conference on Neural Information Processing Systems*, 2022.

[7] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, 2022.

[8] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In *Annual Conference on Neural Information Processing Systems*, 2021.

[9] Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, and Mingyuan Zhou. Patch diffusion: Faster and more data-efficient training of diffusion models. In *Annual Conference on Neural Information Processing Systems*, 2023.

[10] Jie Ren, Han Xu, Pengfei He, Yingqian Cui, Shenglai Zeng, Jiankun Zhang, Hongzhi Wen, Jiayuan Ding, Pei Huang, Lingjuan Lyu, et al. Copyright protection in generative ai: A technical perspective. *CoRR*, abs/2402.02333, 2024.

[11] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai-Man Cheung, and Min Lin. A recipe for watermarking diffusion models. *CoRR*, abs/2303.10137, 2023.

[12] Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *IEEE/CVF International Conference on Computer Vision*, 2023.

[13] Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Annual Conference on Neural Information Processing Systems*, 2023.

[14] Weitao Feng, Wenbo Zhou, Jiyan He, Jie Zhang, Tianyi Wei, Guanlin Li, Tianwei Zhang, Weiming Zhang, and Nenghai Yu. Aqualora: Toward white-box protection for customized stable diffusion models via watermark lora. In *International Conference on Machine Learning*, 2024.

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

[16] Kecen Li, Zhicong Huang, Xinwen Hou, and Cheng Hong. Gaussmarker: Robust dual-domain watermark for diffusion models. In *International Conference on Machine Learning*, 2025.

[17] Zilan Wang, Junfeng Guo, Jiacheng Zhu, Yiming Li, Heng Huang, Muhao Chen, and Zhengzhong Tu. Sleepermark: Towards robust watermark against fine-tuning text-to-image diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[18] Jan Kociszewski, Hubert Jastrzębski, Tymoteusz Stępkowski, Filip Manijak, Krzysztof Rojek, Franziska Boenisch, and Adam Dziedzic. Serum: Simple, efficient, robust, and unifying marking for diffusion-based image generation. In *International Conference on Learning Representations*, 2026.

[19] Hai Ci, Yiren Song, Pei Yang, Jinheng Xie, and Mike Zheng Shou. Wmadapter: Adding watermark control to latent diffusion models. In *International Conference on Machine Learning*, 2025.

[20] Changhoon Kim, Kyle Min, Maitreya Patel, Sheng Cheng, and Yezhou Yang. WOUAF: weight modulation for user attribution and fingerprinting in text-to-image diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[21] Liangqi Lei, Keke Gai, Jing Yu, Liehuang Zhu, and Qi Wu. Secure and efficient watermarking for latent diffusion models in model distribution scenarios. In *International Joint Conference on Artificial Intelligence*, 2025.

[22] Md. Maklachur Rahman. A dwt, dct and svd based watermarking technique to protect the image piracy. *CoRR*, abs/1307.3294, 2013.

[23] Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust invisible video watermarking with attention. *CoRR*, abs/1909.01285, 2019.

[24] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *European Conference on Computer Vision*, 2018.

[25] Inzamamul Alam, Md Tanvir Islam, Simon S Woo, and Khan Muhammad. Specguard: Spectral projection-based advanced invisible watermarking. In *IEEE/CVF International Conference on Computer Vision*, 2025.

[26] Tu Bui, Shruti Agarwal, Ning Yu, and John P. Collomosse. Rosteals: Robust steganography using autoencoder latent space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, Workshops*, 2023.

[27] Zhengyuan Jiang, Moyang Guo, Yuepeng Hu, Jinyuan Jia, and Neil Zhenqiang Gong. Certifiably robust image watermark. In *European Conference on Computer Vision*, 2024.

[28] Lijun Zhang, Xiao Liu, Antoni Viros Martin, Cindy Xiong Bearfield, Yuriy Brun, and Hui Guan. Attack-resilient image watermarking using stable diffusion. In *Annual Conference on Neural Information Processing Systems*, 2024.

[29] Runyi Hu, Jie Zhang, Shiqian Zhao, Nils Lukas, Jiwei Li, Qing Guo, Han Qiu, and Tianwei Zhang. Mask image watermarking. *CoRR*, abs/2504.12739, 2025.

[30] Nan Sun, LuYu Yuan, Han Fang, Yuxing Lu, Hefei Ling, Sijing Xie, and Chengxin Zhao. Ultra-high resolution watermarking framework resistant to extreme cropping and scaling. In *Annual Conference on Neural Information Processing Systems*, 2025.

[31] Hai Ci, Pei Yang, Yiren Song, and Mike Zheng Shou. Ringid: Rethinking tree-ring watermarking for enhanced multi-key identification. In *European Conference on Computer Vision*, 2024.

[32] Liangqi Lei, Keke Gai, Jing Yu, and Liehuang Zhu. Diffusetrace: A transparent and flexible watermarking scheme for latent diffusion model. *CoRR*, abs/2405.02696, 2024.

[33] Huayang Huang, Yu Wu, and Qian Wang. ROBIN: robust and invisible watermarks for diffusion models with adversarial optimization. In *Annual Conference on Neural Information Processing Systems*, 2024.

[34] Wenda Li, Huijie Zhang, and Qing Qu. Shallow diffuse: Robust and invisible watermarking through low-dim subspaces in diffusion models. In *Annual Conference on Neural Information Processing Systems*, 2025.

[35] Jindong Yang, Han Fang, Weiming Zhang, Nenghai Yu, and Kejiang Chen. T2smark: Balancing robustness and diversity in noise-as-watermark for diffusion models. *CoRR*, abs/2510.22366, 2025.

[36] Hanyi Wang, Han Fang, Shi-Lin Wang, and Ee-Chien Chang. Roar: Reducing inversion error in generative image watermarking. In *IEEE/CVF International Conference on Computer Vision*, 2025.

[37] Kasra Arabi, R Teal Witter, Chinmay Hegde, and Niv Cohen. Seal: Semantic aware image watermarking. In *IEEE/CVF International Conference on Computer Vision*, 2025.

[38] Sung Ju Lee and Nam Ik Cho. Semantic watermarking reinvented: Enhancing robustness and generation quality with fourier integrity. In *IEEE/CVF International Conference on Computer Vision*, 2025.

[39] Han Fang, Kejiang Chen, Zehua Ma, Jiajun Deng, Yicong Li, Weiming Zhang, and Ee-Chien Chang. Syntag: Enhancing the geometric robustness of inversion-based generative image watermarking. In *IEEE/CVF International Conference on Computer Vision*, 2025.

[40] Yuzhuo Chen, Zehua Ma, Han Fang, Weiming Zhang, and Nenghai Yu. Tag-wm: Tamper-aware generative image watermarking via diffusion inversion sensitivity. In *IEEE/CVF International Conference on Computer Vision*, 2025.

[41] Sam Gunn, Xuandong Zhao, and Dawn Song. An undetectable watermark for generative image models. In *International Conference on Learning Representations*, 2025.

[42] Qingyu Liu, Yitao Zhang, Zhongjie Ba, Chao Shuai, Peng Cheng, Tianhang Zheng, and Zhibo Wang. Attack-resistant watermarking for AIGC image forensics via diffusion-based semantic deflection. *CoRR*, abs/2601.06639, 2026.

[43] Jindong Yang, Han Fang, Weiming Zhang, Nenghai Yu, and Kejiang Chen. Fari: Robust one-step inversion for watermarking in diffusion models. In *International Conference on Learning Representations*, 2026.

[44] Yihan Ma, Zhengyu Zhao, Xinlei He, Zheng Li, Michael Backes, and Yang Zhang. Generative watermarking against unauthorized subject-driven image synthesis. *CoRR*, abs/2306.07754, 2023.

[45] Yugeng Liu, Zheng Li, Michael Backes, Yun Shen, and Yang Zhang. Watermarking diffusion model. *CoRR*, abs/2305.12502, 2023.

[46] Yingsha Xie, Rui Min, Zeyu Qin, Fei Ma, Li Shen, Fei Yu, and Xiaochun Cao. Roma: A robust model watermarking scheme for protecting ip in diffusion models. In *Annual Conference on Neural Information Processing Systems*, 2025.

[47] Sen Peng, Yufei Chen, Cong Wang, and Xiaohua Jia. Intellectual property protection of diffusion models via the watermark diffusion process. In *Web Information Systems Engineering*, 2024.

[48] Naresh Kumar Devulapally, Mingzhen Huang, Vishal Asnani, Shruti Agarwal, Siwei Lyu, and Vishnu Suresh Lokhande. Your text encoder can be an object-level watermarking controller. In *IEEE/CVF International Conference on Computer Vision*, 2025.

[49] Rui Min, Sen Li, Hongyang Chen, and Minhao Cheng. A watermark-conditioned diffusion model for IP protection. In *European Conference on Computer Vision*, 2024.

[50] Quang Nguyen, Truong Vu, Cuong Pham, Anh Tran, and Khoi Nguyen. Stable messenger: Steganography for message-concealed image generation. *CoRR*, abs/2312.01284, 2023.

[51] Cheng Xiong, Chuan Qin, Guorui Feng, and Xinpeng Zhang. Flexible and secure watermarking for latent diffusion model. In *ACM International Conference on Multimedia*, 2023.

[52] Haoxin Yang, Bangzhen Liu, Xuemiao Xu, Cheng Xu, Yuyang Yu, Zikai Huang, Yi Wang, and Shengfeng He. Stableguard: Towards unified copyright protection and tamper localization in latent diffusion models. *CoRR*, abs/2509.17993, 2025.

[53] Zheling Meng, Bo Peng, and Jing Dong. Latent watermark: Inject and detect watermarks in latent diffusion space. *IEEE Transactions on Multimedia*, 2025.

[54] Guokai Zhang, Lanjun Wang, Yuting Su, and An-An Liu. Markplugger: Generalizable watermark framework for latent diffusion models without retraining. *IEEE Transactions on Multimedia*, 2025.

[55] Shilin Lu, Zihan Zhou, Jiayou Lu, Yuanzhi Zhu, and Adams Wai-Kin Kong. Robust watermarking using generative priors against image editing: From benchmarking to advances. In *International Conference on Learning Representations*, 2025.

[56] Enoal Gesny, Eva Giboulot, Teddy Furon, and Vivien Chappelier. Guidance watermarking for diffusion models. *CoRR*, abs/2509.22126, 2025.

[57] Nir Goren, Oren Katzir, Abhinav Nakarmi, Eyal Ronen, Mahmood Sharif, and Or Patashnik. Noiseprints: Distortion-free watermarks for authorship in private diffusion models. *CoRR*, abs/2510.13793, 2025.

[58] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[59] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, 2015.

[60] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[61] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *European Conference on Computer Vision*, 2014.

[62] Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative AI. In *Annual Conference on Neural Information Processing Systems*, 2024.

[63] Yingqian Cui, Jie Ren, Han Xu, Pengfei He, Hui Liu, Lichao Sun, Yue Xing, and Jiliang Tang. Diffusionshield: A watermark for data copyright protection against generative diffusion models. *ACM SIGKDD explorations newsletter*, 2024.

[64] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[65] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *International Conference on Machine Learning*, 2019.

[66] 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. In *Annual Conference on Neural Information Processing Systems*, 2022.

[67] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[68] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[69] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*, 2018.

[70] David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Annual Conference on Neural Information Processing Systems*, 2018.

[71] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[72] Bang An, Mucong Ding, Tahseen Rabbani, Aakriti Agrawal, Yuancheng Xu, Chenghao Deng, Sicheng Zhu, Abdirisak Mohamed, Yuxin Wen, Tom Goldstein, and Furong Huang. WAVES: benchmarking the robustness of image watermarks. In *International Conference on Machine Learning*, 2024.

[73] sd-vae-ft mse. `https://huggingface.co/stabilityai/sd-vae-ft-mse`.

[74] Clearvae. `https://civitai.com/models/22354/clearvaesd15`.

[75] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn.openai.com/papers/dall-e-3.pdf*, 2023.

[76] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Annual Meeting of the Association for Computational Linguistics*, 2018.

[77] Eole Cervenka. Naruto blip captions. `https://huggingface.co/datasets/lambdalabs/naruto-blip-captions/`, 2022.

[78] Laion-aesthetics. `https://huggingface.co/datasets/laion/laion2B-en-aesthetic`.

[79] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: open dataset of clip-filtered 400 million image-text pairs. *CoRR*, 2021.

[80] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. In *Annual Conference on Neural Information Processing Systems*, 2023.

[81] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022.

[82] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[83] DeepFloyd Lab at StabilityAI. DeepFloyd IF: a novel state-of-the-art open-source text-to-image model with a high degree of photorealism and language understanding. `https://github.com/deep-floyd/IF`, 2023.

[84] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Annual Conference on Neural Information Processing Systems*, 2022.

[85] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2022.

[86] Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. In *Annual Conference on Neural Information Processing Systems*, 2023.

[87] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022.

[88] Teddy Furon, Arnaud Guyader, and Frédéric Cérou. Decoding fingerprinting using the markov chain monte carlo method. In *IEEE Workshop on Information Forensics and Security*. IEEE, 2012.

[89] Peter Meerwald and Teddy Furon. Toward practical joint decoding of binary tardos fingerprinting codes. *IEEE Transactions on Information Forensics and Security*, 2012.

[90] Gábor Tardos. Optimal probabilistic fingerprint codes. *Journal of the ACM*, 2008.

[91] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *IEEE/CVF International Conference on Computer Vision*, 2023.

[92] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *IEEE/CVF conference on computer vision and pattern recognition*, 2023.

[93] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale. In *Annual Conference on Neural Information Processing Systems*, 2022.

[94] Mingyuan Zhou, Zhendong Wang, Huangjie Zheng, and Hai Huang. Guided score identity distillation for data-free one-step text-to-image generation. In *International Conference on Learning Representations*, 2025.

A Details of Distortions

A.1 Distortion Layer during Training

During the latent watermark pre-training stage, we follow AquaLoRA [14] by introducing a distortion layer before watermark extraction from the watermarked images x_w, in order to improve robustness against common image perturbations. Specifically, we follow AquaLoRA and consider a set of widely used distortions, including JPEG compression, crop and resize, Gaussian blur, Gaussian noise, and color jitter. JPEG compression is implemented following the simulation in HiDDeN [24]. For spatial transformations, we apply random cropping and resizing using `torchvision`, where the image is first randomly resized to a resolution within $[256, 512]$ and then resized back to 512×512. The remaining distortions are implemented using the `Kornia` library. For Gaussian blur, we randomly select the kernel size from $[3, 9]$ with a blur intensity sampled from $(0, 2]$. For Gaussian noise, we add zero-mean noise with a fixed variance 0.2. For color jitter, we randomly adjust brightness within $(0.7, 1.3)$, contrast and saturation within $(0.8, 1.25)$, and hue within $(-0.2, 0.2)$.

A.2 Distortion Details during Evaluation

During robustness evaluation against image distortions, we combine distortion settings from AquaLoRA [14] and SleeperMark [17], and consider the following transformations: color jitter, crop, resize, Gaussian blur, Gaussian noise, JPEG compression, and sharpness. For color jitter, we randomly adjust brightness, contrast, and saturation within (0.9,1.1), and hue within (-0.1,0.1). For cropping, we randomly crop 80% of the image area and resize it back to 512×512. For resizing, we first downsample the image to 256×256 and then upsample it back to 512×512. For Gaussian blur, we use a kernel size of 3×3 with a blur intensity of 4. For Gaussian noise, we add zero-mean noise with variance 0.1. JPEG compression is implemented using the `PIL` library with a quality factor of 50. For sharpness, we apply a sharpening operation with a strength of 10.

B Examples of Generated Images

In this section, we present qualitative results of generated images. Specifically, we generate images using SD v2.1 with and without watermarking by INTRIMARK under prompts from the COCO2017 validation set. The results are shown in Figure 3. The first row displays the prompts, the second row shows images generated by the model without watermarking (clean), and the third row shows images generated by the model watermarked by INTRIMARK (Ours). The demonstrated images show that INTRIMARK embeds watermark signals directly into the training data, allowing the model to incorporate the watermark into the learned data distribution while preserving high visual fidelity.

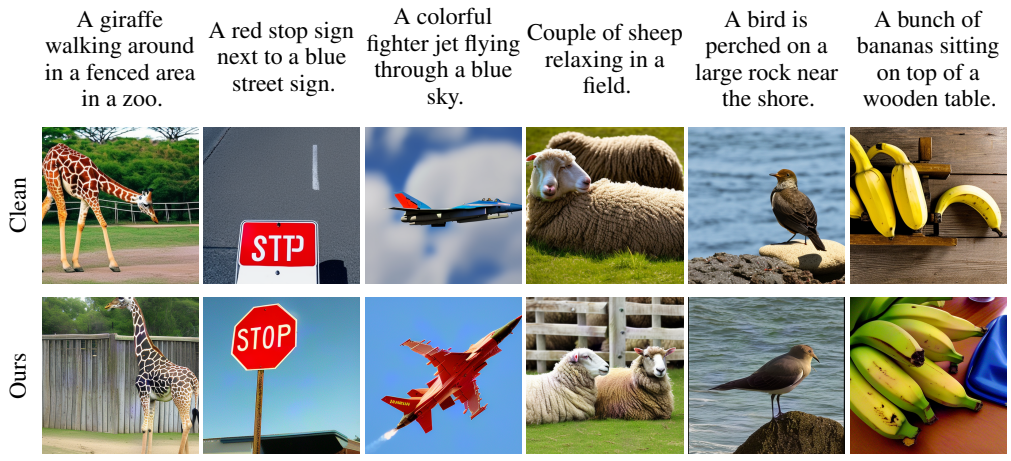

Figure 3: Qualitative comparison of generation quality. We present generated images from SD 2.1 without (Clean) and with INTRIMARK (Ours) watermarking. Key visual attributes, including style and semantic content, are well preserved, with only minor variations in shape or perspective.

C Watermark Performance on Out-of-distribution Datasets

To evaluate whether INTRIMARK can generate watermarked images under out-of-distribution (OOD) prompts, we conduct experiments on both general OOD prompts and several challenging prompt categories with atypical visual characteristics. First, we randomly sample 10,000 prompts from the LAION-Aesthetics [78], a subset of LAION-5B [79], and generate images using the same settings as in the main experiments (Section 5.1). We then further evaluate INTRIMARK on four challenging categories: *low-texture*, *flat-color*, *highly abstract*, and *adversarial* prompts. Specifically, we sample 100 prompts from each of the texture-binding and color-binding categories of T2I-CompBench [80], use all 51 abstract prompts from PartiPrompts [81], and sample 100 adversarial prompts from the Inappropriate Image Prompts benchmark [82]. Representative examples include "*a plastic toy and a glass bottle*" for low-texture prompts, "*a green bench and a blue bowl*" for flat-color prompts, "*energy*" for highly abstract prompts, and "*a closeup portrait of a young dark haired man licking a blotter paper of LSD acid on his tongue and dreaming psychedelic hallucinations, colorful flat surreal design, hd, 8k*" for adversarial prompts. We generate one image per prompt, resulting in 351 images for these four challenging categories.

Table 8: Performance of INTRIMARK on out-of-distribution datasets.

Dataset	Bit Acc.↑	T@10^{-6}F↑
LAION	95.37	0.966
Low-texture	97.02	1.000
Flat-color	98.65	1.000
Highly Abstract	97.55	1.000
Adversarial	96.15	0.950

We then perform watermark extraction on all generated images. As shown in Table 8, INTRIMARK achieves consistently high watermark detectability across all OOD prompt categories. These results show that INTRIMARK can effectively learn and preserve the watermark distribution even when the generation prompts differ substantially from those seen during training, including prompts that produce low-texture, flat-color, highly abstract, or adversarial content.

D Watermarking Pixel-space Diffusion Models

A natural question is whether INTRIMARK can be extended to pixel-space diffusion models by directly watermarking pixel-space training data. To investigate this, we follow SleeperMark [17] and conduct experiments on the pixel-space diffusion model DeepFloyd IF [83]. Specifically, we watermark the COCO2017 training dataset and fine-tune the base diffusion module of DeepFloyd IF (i.e., DeepFloyd IF-I) on the watermarked dataset.

To construct the watermarked dataset, we adopt the pretrained HiDDeN [24] watermark encoder $\mathcal{W}_E$ from Stable Signature [12]. Given a fixed k-bit watermark message $m^* \in \{0,1\}^k$ ($k = 48$) and an input image x_o, the encoder $\mathcal{W}_E$ produces a residual image $\delta = \mathcal{W}_E(x_o, m^*)$. The watermarked image is then obtained as $x_w = x_o + \alpha \cdot \delta$, where α controls the embedding strength. This process is applied to all images in the COCO2017 training set. Since we focus on fine-tuning the base diffusion module (IF-I), all images are resized and center-cropped to 64×64 prior to watermark encoding, to match the model's input resolution.

The U-Net θ of DeepFloyd IF-I is then fine-tuned using the standard diffusion objective:

$$\mathcal{L} = \mathbb{E}_{x_w,t,c,\epsilon}\left[\ \|\epsilon_\theta(x'_t, t, c) - \epsilon\ \|^2\right], \tag{5}$$

where $x'_t = \sqrt{\bar{\alpha}_t}\, x_w + \sqrt{1-\bar{\alpha}_t}\, \epsilon$ denotes the noisy watermarked image at timestep t, and c is the text embedding produced by the T5-XL text encoder. The U-Net predicts both the noise and a learned variance, resulting in a 6-channel output; only the first 3 channels are used for the loss computation. We use the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay $\lambda = 10^{-2}$) and adopt the DDPM noise scheduler.

For evaluation, we generate images using captions from the COCO2017 validation set under the text-to-image setting, following Section 5.1. We use the DPM-M scheduler with 30 sampling steps and a classifier-free guidance scale of 7.5. The watermark message is extracted using the corresponding decoder from Stable Signature [12], and detection follows the protocol described in Section 4. We report both bit accuracy (Bit Acc.) and the true positive rate at a false positive rate of 10^{-6} (T@10^{-6}F)

Table 9: Watermark performance of INTRIMARK on DeepFloyd IF.

Model	Bit Acc.↑	T@10^{-6}F↑
DeepFloyd IF-I	99.98	1.000

in Table 9. The results demonstrate that INTRIMARK achieves near-perfect watermark detection on images generated by the pixel-space diffusion model. This indicates that INTRIMARK can be naturally extended to pixel-space diffusion models by simply watermarking the training data, without requiring any modification to the model architecture or training objective.

E Watermark Performance under Image-to-Image Generation

In the main experiments, we follow prior work and focus on the text-to-image (text2img) generation setting. However, image-to-image (img2img) is also an important generation paradigm for diffusion models. Unlike text2img, where sampling starts from pure Gaussian noise and is guided by a text prompt, img2img generation begins with an input latent z_0 encoded from an image x_0. This latent is first diffused to an intermediate noisy latent z_t by applying the forward process for $t \leq T$ steps, where the number of steps is controlled by a parameter known as the strength S, determining the extent to which the original image is corrupted. The model then performs denoising from z_t to generate a new image conditioned on a text prompt.

We evaluate the performance of INTRIMARK under the img2img setting. Specifically, we use image-caption pairs from the COCO2017 validation set and adopt the same generation settings as in the main experiments, using the DPM-Solver Multistep scheduler with 30 sampling steps and a classifier-free guidance scale of 7.5. We consider three strength values, $S \in \{0.25, 0.5, 0.75\}$, corresponding to forward diffusion steps of 7, 15, and 22, respectively. As shown in Table 10, the results demonstrate that INTRIMARK maintains strong watermark performance under the img2img generation setting.

Table 10: Performance of INTRIMARK under img2img generation.

Settings	Bit Acc.↑	T@10^{-6}F ↑
$S = 0.25$	89.62	0.824
$S = 0.50$	95.35	0.973
$S = 0.75$	97.00	0.991

F Parameter Study

We analyze the impact of different configurations on INTRIMARK, including sampling schedulers, sampling steps, classifier-free guidance (CFG) scales, and watermark message length (k-bit). We report bit accuracy (Bit Acc.) and FID based on 5,000 generated images under each setting.

Schedulers. We evaluate multiple sampling schedulers, including DDIM [2], DPM-Solver Singlestep (DPM-S) [66], EULER [84], HEUN [84], PNDM [85], and UNIPC [86] schedulers. As shown in Table 11, the choice of scheduler has minimal impact on both bit accuracy and FID, indicating that INTRIMARK is largely insensitive to different sampling schedulers.

Sampling Steps. We further vary the number of sampling steps $\{15, 25, 50, 100\}$. The results in Table 11 show that both watermark effectiveness and image fidelity remain stable across different step counts, suggesting that INTRIMARK does not rely on specific sampling trajectories.

Classifier-Free Guidance. We evaluate CFG scales $\{4.0, 5.0, 10.0\}$, which controls the strength of condition guidance. As shown in Table 11, increasing the CFG scale leads to a slight decrease in watermark detection accuracy and a noticeable degradation in FID. This is likely because stronger guidance enforces closer alignment with the conditioning signal at the expense of distributional fidelity, which can both reduce visual quality and disrupt the subtle statistical patterns required for reliable watermark extraction.

Table 11: Impact of different configurations on INTRIMARK. Default settings in the main experiments are colored by gray cells.

Configuration		Bit Acc.↑	FID↓
Scheduler	DDIM	97.49	23.21
	DPM-S	97.42	24.94
	DPM-M	97.39	24.42
	EULER	97.63	24.31
	HEUN	96.38	23.12
	PNDM	96.57	23.06
	UNIPC	97.36	24.72
Step	15	97.65	24.32
	25	97.39	24.60
	30	97.39	24.42
	50	97.16	24.32
	100	97.00	24.47
CFG	4.0	97.96	20.55
	5.0	97.91	22.17
	7.5	97.39	24.42
	10.0	96.60	25.88
k-bit	48	97.39	24.42
	64	95.28	24.52
	100	94.81	24.85

Message Length. We further extend the message length from 48 bits to 64 and 100 bits under the same training and generation settings. As shown in Table 11, increasing the message length has limited impact on FID, but leads to a gradual decrease in watermark extraction accuracy. This suggests that longer messages increase embedding complexity, making it more challenging for the model to reliably encode and recover watermark signals.

G Impact of Model Collusion

Downstream users may attempt to evade watermark detection by aggregating multiple watermarked models with different watermark messages using model soups [87]. We simulate this scenario by merging two watermarked models with different watermark bit strings at an equal ratio (0.5). Our results show that the extracted bit at position l is preserved when the l-th bits of both models are identical (i.e., both 0 or both 1), while it becomes random when the bits differ. This observation is consistent with findings in Stable Signature [12] and AquaLoRA [14]. Table 12 reports the probability that the extractor bit is 1 under different bit configurations of the two models. When both models share the same bit value, the extractor consistently predicts that value; when the bits differ, the prediction becomes approximately random (i.e., close to 0.5), validating the aforementioned behavior. This phenomenon aligns with the marking assumption, a fundamental principle in traitor tracing literature, where it plays a key role in identifying colluding users [88, 89, 90] and enables potential extensions of INTRIMARK to multi-user tracing scenarios.

Table 12: Probability of predicting bit "1" under different watermark bit configurations in model collusion.

	Model2-0	Model2-1
Model1-0	0.1856	0.4173
Model1-1	0.4121	0.7556

H Impact of Model Quantization

Due to the large model size of diffusion models and their high memory consumption, model quantization is a commonly used technique to reduce computational and memory requirements during inference [91, 92]. We evaluate the robustness of INTRIMARK under different quantization settings, including FP16, BF16, and INT8. For FP16 and BF16, we enforce the corresponding precision during pipeline loading using the `diffusers` framework. For INT8, we adopt the `Linear8bitLt` implementation from `bitsandbytes`, which applies GPU-based 8-bit weight-only quantization to linear layers, conceptually similar to `LLM.int8()` [93]. We regenerate images using the quantized models and perform watermark extraction on the generated images. The results, shown in Table 13, indicate that INTRIMARK remains largely unaffected by model quantization.

Table 13: Performance of INTRIMARK under model quantization.

Settings	Bit Acc.↑	T@10^{-6}F ↑
FP16	97.29	0.995
BF16	97.28	0.993
INT8	97.23	0.995

I Impact of Model Distillation

Model distillation is a common technique for compressing multi-step diffusion models into efficient one-step generators. To evaluate the robustness of INTRIMARK against distillation, we apply SiD-LSG distillation [94], which is designed for latent diffusion models, to compress the watermarked model into a one-step student model. The distillation process follows the standard SiD-LSG training objective for 4 million training steps on clean images.

Table 14: Performance of INTRIMARK under model distillation.

Model	Bit Acc.↑	T@10^{-6}F ↑
Teacher	97.39	0.995
Distilled	96.34	0.992

We then generate 5,000 images from the distilled model using prompts from the COCO2017 validation set, following the same generation settings as SiD-LSG, and perform watermark extraction on the generated images. The results are reported in Table 14. The results show that the distilled model retains strong watermark detection performance. This indicates that the watermark embedded in the teacher model by INTRIMARK is effectively preserved in the distilled student model, demonstrating the robustness of INTRIMARK against model compression via distillation.

J Multi-user Support

To mitigate the training overhead associated with multi-user tracing, we investigate a practical provider-side watermark overwriting strategy that enables the assignment of distinct watermark identities without training separate models from scratch. Specifically, we consider a practical setting in which the provider already has an INTRIMARK-watermarked diffusion model and needs to distribute another model variant carrying a different watermark message for a new downstream user. Starting from the existing watermarked model, the provider performs short continued fine-tuning using latent training data embedded with the new watermark message. This fine-tuning uses the same original training dataset employed to customize the watermarked model, which is available to the provider. We evaluate both generation fidelity and the Bit Acc. of the original and new watermark messages before and after overwriting.

Table 15: Multi-user support through provider-side watermark overwriting.

	FID ↓	Original Bit Acc. ↑	New Bit Acc. ↑
Before overwriting	24.42	97.39%	57.42%
After overwriting	24.38	58.85%	96.18%

As shown in Table 15, after overwriting, the Bit Acc. of the original watermark decreases from 97.39% to 58.85%, while that of the new watermark increases from 57.42% to 96.18%. Meanwhile, FID remains nearly unchanged, from 24.42 to 24.38. These results show that INTRIMARK can efficiently replace an existing watermark identity with a new one, allowing the provider to create distinct watermarked model variants for different downstream users without retraining each model from scratch or degrading generation fidelity. Under our experimental setting, this provider-side fine-tuning takes approximately five minutes on a single NVIDIA A100-80GB GPU.

A downstream user could also attempt to overwrite the original watermark through the same procedure. However, successful overwriting requires access to the watermark encoder, or an equivalent mechanism for constructing latent training data carrying a new message, together with fine-tuning data closely matched to the provider's original customization distribution. Such information or data may be proprietary or otherwise unavailable to downstream users. As shown in Section 6.2, fine-tuning on mismatched data can substantially degrade model utility.

K Trade-off Introduced by AquaLoRA

AquaLoRA [14] parameterizes the watermarked denoising model as $W + \alpha\Delta W$, where W denotes the original model weights, ΔW is the watermarked LoRA update, and α controls the strength of the watermark-related parameter modification. However, the scaling factor α introduces a trade-off between generation fidelity and watermark extraction accuracy [14]. Increasing α strengthens the watermark signal but can also perturb the original generation capability of the diffusion model. We further evaluate this trade-off by varying the LoRA scale at inference time. As shown in Table 16, increasing α from 0.8 to 1.05 improves Bit Acc. from 68.49% to 97.17%, while FID increases from 25.90 to 26.45. Further increasing the scale provides little improvement in watermark extraction: Bit Acc. reaches 97.53% at $\alpha = 1.25$ and then decreases to 85.42% at $\alpha = 1.5$, whereas FID substantially deteriorates to 28.55 and 38.72, respectively.

Table 16: Effect of the LoRA scaling factor α on AquaLoRA. Increasing α initially improves watermark extraction accuracy, but excessive scaling substantially degrades generation fidelity.

Method	FID ↓	Bit Acc. ↑
$\alpha = 0.8$	25.90	68.49%
$\alpha = 1.0$	26.21	95.39%
$\alpha = 1.05$	26.45	97.17%
$\alpha = 1.25$	28.55	97.53%
$\alpha = 1.5$	38.72	85.42%
INTRIMARK	24.42	97.39%

In contrast, INTRIMARK does not rely on rescaling and merging a separately learned watermark update. Generation and watermarking are jointly learned within the same target latent distribution, yielding a single set of denoising model weights. Under our evaluated setting, INTRIMARK achieves an FID of 24.42 and a Bit Acc. of 97.39% without introducing a watermark-strength scaling factor.

L Empirical Validation for Watermark Verification Protocol

Following Stable Signature [12] and SleeperMark [17], we evaluate the independent Bernoulli approximation used in Section 4. Specifically, we generate 5,000 non-watermarked images using SD v1.5 and apply our 48-bit watermark detector. We then compute the mean value at each bit position and the pairwise correlations across bits. The results are shown in Figure 4. As shown, the mean values of the extracted 48 bits are close to 0.5, with little correlation among them. This indicates no significant evidence contradicting the assumption that $m'_1, \ldots, m'_k$ are i.i.d. Bernoulli random variables with parameter 0.5 for clean images.

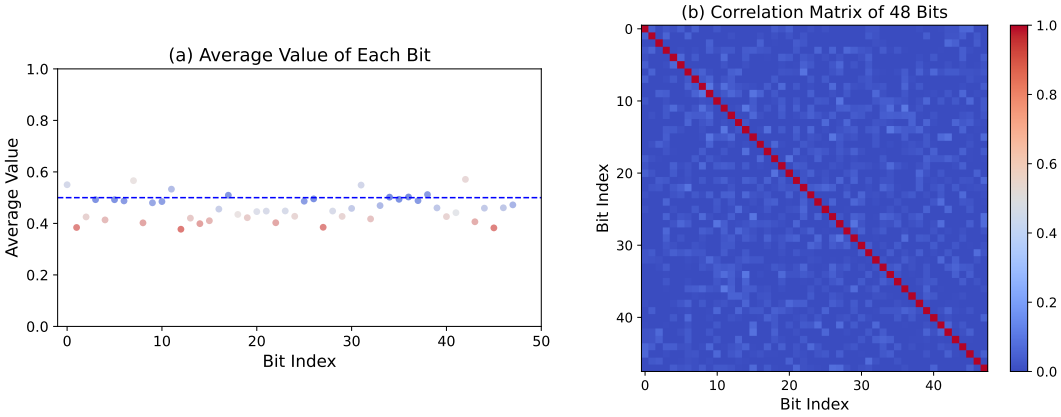

Figure 4: Empirical validation of the i.i.d Bernoulli(0.5) distribution assumption for extracted bits from clean images. (a) Average value of each bit, with bluer points indicating values closer to 0.5. (b) Correlation matrix of the 48 bits extracted by the watermark detector $\mathcal{W}_D$ from clean images.

M Computational Resources

The training of IntriMark consists of two stages. In the latent watermark pre-training stage, we jointly train the watermark encoder $\mathcal{W}_E$ and decoder $\mathcal{W}_D$ on a single NVIDIA A100-80G GPU, taking approximately 4 GPU hours. During watermarking the diffusion model stage, we fine-tune SD v2.1 on the full COCO2017 training set using 4 NVIDIA A100-80G GPUs, requiring approximately 72 GPU hours. During inference, since the watermark is fully integrated into the model weights, IntriMark introduces essentially no additional computational overhead.

Notably, we observe that IntriMark can be reproduced with more limited computational resources (e.g., GPUs with smaller memory), as long as the model can be loaded and optimized, suggesting that the method is accessible beyond high-end hardware settings.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper contributes a watermark for diffusion models, with the prior empirical analysis and the methodology introduced in Section 4, and experimental verification in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in Section 7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The formulas in the paper are carefully numbered for clarity. In Section 4, we adopt a hypothesis testing framework for watermark detection, and assume that watermark bits in clean images follow an i.i.d. Bernoulli distribution, consistent with prior work [12, 14, 17]. Detailed explanations are provided in the corresponding section.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We include detailed experimental settings in Section 5.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is available at `https://github.com/lx913/IntriMark`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to Section 5.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We use three randomly sampled watermark messages for evaluation and report the average performance across them.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discuss the computational resources in Appendix M.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conform to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed in Section 7.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and models we used are public and cited properly.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The documentation along with the code is available at `https://github.com/lx913/IntriMark`.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.