Transferable Multi-Bit Watermarking Across Frozen Diffusion Models via Latent Consistency Bridges

Hong-Hanh Nguyen-Le [1 *] **Van-Tuan Tran** [2 *] **Thuc D. Nguyen** [3] **Nhien-An Le-Khac** [1]

Abstract

As generative AI advances, global governance frameworks increasingly mandate verifiable content provenance. However, existing watermarking techniques face a critical policy-to-technology disconnect: sampling-based methods require computationally prohibitive inversion, while fine-tuning approaches are tethered to specific model checkpoints, hindering standardized, cross-model oversight. To bridge this gap, we introduce DiffMark, a plug-and-play multi-bit watermarking framework. DiffMark embeds a persistent, learned perturbation into every denoising step of a frozen diffusion model, accumulating a recoverable signal in the final latent space. To enable efficient training through the frozen network, we utilize Latent Consistency Models (LCMs) as a differentiable training bridge. DiffMark achieves 64-bit extraction in a single 16.4 ms forward pass, which is a $45\times$ speed-up over inversion baselines. By enabling per-image key flexibility and cross-architecture transferability without retraining, DiffMark provides the practical, scalable technical tooling necessary to operationalize user accountability and enforce emerging AI governance mandates.

1. Introduction

As advanced AI systems are developed and deployed, governments and policymakers are increasingly recognizing the need to take targeted action to access the benefits and mitigate the risks of AI. The past three years have seen the establishment of new AI governance institutes in the US, UK, China, Korea, and across the world (UK Department for Science, Innovation and Technology, 2023; U.S. National Institute of Standards and Technology (NIST), 2023; AI Seoul Summit, 2024; China AI Safety and Development Association, 2025); the emergence of a private third-party AI evaluation ecosystem; and the passage of prominent governance laws and frameworks for general-purpose AI systems (e.g., the EU AI Act (European Parliament and Council of the European Union, 2024a), California SB-53 (California State Legislature, 2025), NY RAISE Act (New York State Legislature, 2025), and China's "Basic security requirements for generative artificial intelligence services" standard (National Technical Committee 260 on Cybersecurity of Standardization Administration of China (SAC/TC260), 2024)). Regulatory and oversight frameworks are increasingly reliant on rigorous techniques for transparency, monitoring, and verification. However, there are significant disconnects between policy aspirations and the technical tooling necessary for their realization (Reuel et al., 2024).

A pressing example of this policy-to-technology disconnect lies in the governance of generative AI. The rapid proliferation of diffusion models (DMs) (Rombach et al., 2022; Podell et al., 2024; Ruiz et al., 2023; Esser et al., 2024) has enabled photorealistic image creation at an unprecedented scale. Despite practical applications in entertainment and science discovery, these models escalate real-world risks of deepfakes, misinformation, and copyright infringement (Pearson & Zinets, 2022; Pezenik & Shepherd, 2024; Tenbarge, 2024; CNN, 2024). In response to these challenges and to fulfill the requirements of emerging mandates like the EU AI Act (European Parliament and Council of the European Union, 2024b) and C2PA (Coalition for Content Provenance and Authenticity, 2024), watermarking has emerged as a critical provenance technique for embedding imperceptible yet recoverable signals into generated images to establish accountability (Nguyen-Le et al., 2025).

Despite this critical role, existing watermarking methods for DMs face limitations that hinder the practical implementation of these governance mandates at a global scale. First, *sampling-based methods* (Wen et al., 2023; Ci et al., 2024; Li et al., 2025) embed watermark information into the initial noise vector z_T or intermediate latent variables, and recover it via Denoising Diffusion Implicit Models (DDIM) inversion (Song et al., 2020). Despite enabling plug-and-play

[*]Equal contribution [1]School of Computer Science, University College Dublin, Ireland [2]School of Computer Science and Statistics, Trinity College Dublin, Ireland [3]Department of Knowledge Engineering, University of Science, VNU-HCMC, Vietnam. Correspondence to: Hong-Hanh Nguyen-Le <hong-hanh.nguyen-le@ucdconnect.ie>, Nhien-An Le-Khac <an.lekhac@ucd.ie>.

Second Workshop on Technical AI Governance Research (TAIGR) @ ICML 2026, Seoul, South Korea. 2026. Copyright 2026 by the author(s).

deployment, they suffer from three critical drawbacks: (i) detection requires running N-step DDIM inversion (typically $N = 50$), which is costly at platform scale; (ii) most support only zero-bit detection (watermark present or absent), which is impossible for user identification; and (iii) per-image key assignment requires regenerating a new pattern for each image. Second, *fine-tuning-based methods* (Fernandez et al., 2023; Feng et al., 2024; Xiong et al., 2023) enable single-pass multi-bit extraction but fundamentally couple the watermark to a specific model checkpoint. Each new architecture or model variant requires retraining, severely limiting the deployment of standardized, cross-model governance tools across an open-source ecosystem.

To operationalize scalable AI provenance and bridge this technical feasibility gap, we propose ***DiffMark***, a plug-and-play multi-bit watermarking method. Our key insight is that the watermark need not reside in the initial noise z_t, which forces costly inversion for recovery, but can instead be embedded as a persistent learned perturbation δ injected at every denoising step of a frozen DM. A lightweight encoder is used to map an arbitrary L-bit secret to a single latent-space perturbation $\delta \in \mathbb{R}^{4\times h\times w}$, enabling multi-bit capacity and per-image key assignment at inference time. Because δ accumulates throughout the sampling trajectory, its signal is naturally concentrated in the final denoised latent z_0, enabling a lightweight decoder D_ψ to recover the embedded secret in a single forward pass.

Since the entire DM remains completely frozen, the central technical challenge is:

> How can we backpropagate the gradients back to the encoder without traversing the full DDIM denoising chain?

We address this by employing Latent Consistency Models (LCMs) (Luo et al., 2023) as a ***differentiable training bridge***, distilling the multi-step denoising process into $K = 4$ forward passes and providing a tractable gradient path through the frozen UNet. A parallel full-step DDIM path, detached from the encoder graph, supplies the decoder with high-fidelity supervision. This dual-path design decouples two competing requirements: short differentiable paths for encoder learning and realistic high-quality latents for decoder calibration. We further introduce a multi-stage curriculum that prevents optimization collapse by activating reconstruction, imperceptibility, and robustness objectives in strict succession. At inference time, watermarked images are generated with the standard DDIM sampler only at full quality, introducing no runtime overhead. Fig.8 (Appendix) compares the differences between our proposed method and other watermarking paradigms.

Our contributions. In summary, DiffMark addresses the AI governance tooling disconnect by offering key advantages over existing methods:

- **Single-pass Multi-bit Detection**: A lightweight decoder extracts the full L-bit secret directly from the denoised latent z_0 at 16.4 ms, a 45x speedup over sampling-based methods. With $L = 64$ bits, DiffMark provides sufficient capacity for reliable user identification and accountability at a high scale.
- **Key Flexibility**: By leveraging LCMs as a differentiable training bridge, the encoder maps an arbitrary runtime secret to a unique perturbation at inference time, enabling each generated image to carry a distinct tracking key without re-training or fine-tuning.
- **Cross-model Transferability**: The single trained encoder-decoder pair transfers directly across unseen diffusion-based architectures without per-model fine-tuning, supporting standardized evaluation and oversight across a rapidly evolving AI ecosystem.
- **Comparable Robustness**: DiffMark still achieves near-perfect bit accuracy under 13 attack types, including distortion, regeneration, and adversarial attacks, across three datasets: MS-COCO 2017, DiffusionDB, and DALL-E3.

2. Preliminaries

2.1. Latent Diffusion Models

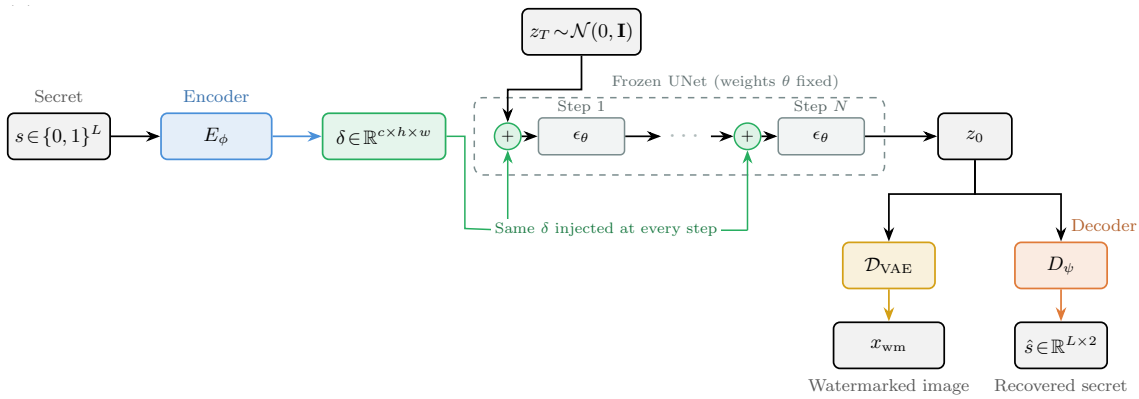
Latent Diffusion Models (LDMs) (Rombach et al., 2022; Podell et al., 2024) operate in a compressed latent space defined by a pretrained VAE: an encoder $\mathcal{E}$ maps an image $x \in \mathbb{R}^{3\times H\times W}$ to a latent representation $z = \mathcal{E}(x) \in \mathbb{R}^{c\times h\times w}$, and a decoder reconstructs $\hat{x} = \mathcal{D}(z)$. A denoising UNet ϵ_θ is trained to reverse a forward process that progressively corrupts a clean latent z_0:

$$z_t = \sqrt{\bar{\alpha}_t}\, z_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \tag{1}$$

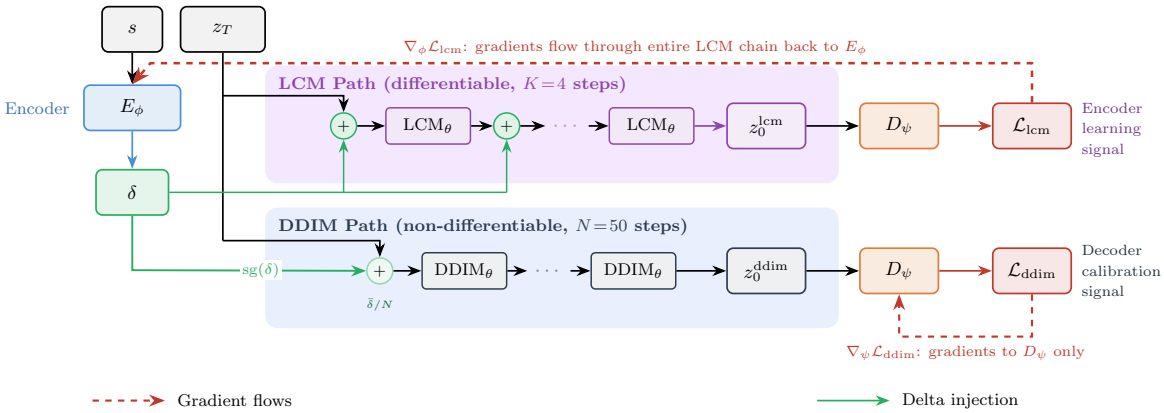
where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. Image generation proceeds by sampling $z_T \sim \mathcal{N}(0, \mathbf{I})$ and iteratively denoising via a sampler such as DDIM (Song et al., 2020):

$$z_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \left(\frac{z_t - \sqrt{1-\bar{\alpha}_t}\, \hat{\epsilon}_t}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1-\bar{\alpha}_{t-1}}\, \hat{\epsilon}_t, \tag{2}$$

where $\hat{\epsilon}_t = \epsilon_\theta(z_t, t, c)$ is the predicted noise. When classifier-free guidance (CFG) (Ho & Salimans, 2022) is used, the noise estimate is replaced by a linear combination of conditional and unconditional predictions: $\hat{\epsilon}_t = \epsilon_{\text{uncond}} + w \cdot (\epsilon_{\text{cond}} - \epsilon_{\text{uncond}})$, where w is the guidance scale. The deterministic nature of DDIM also permits *inversion*: given a clean latent z_0, one can recover an approximate initial noise z_T by reversing Eq.(2), a property exploited by sampling-based watermarking methods (Wen et al., 2023; Ci et al., 2024; Li et al., 2025) for watermark detection.



(a) At inference: a lightweight encoder E_ϕ maps an L-bit secret s to a perturbation δ injected at every denoising step of a frozen UNet. A decoder D_ψ recovers the secret from the final latent z_0 in a single forward pass.



(b) Dual-path training: the differentiable LCM path (K=4 steps) provides encoder gradients via $\mathcal{L}_{\text{lcm}}$, while the full DDIM path (N=50 steps) supplies high-fidelity decoder supervision via $\mathcal{L}_{\text{ddim}}$.

Figure 1. Overview of DiffMark.

2.2. Latent Consistency Models

Latent Consistency Models (LCMs) (Luo et al., 2023) extend consistency distillation (Song et al., 2023) to the latent space of pretrained LDMs. An LCM learns a consistency function $f_\theta : (z_t, \omega, c, t) \mapsto z_0$ that directly predicts the solution of the *augmented Probability Flow Ordinary Differential Equation (PF-ODE)*, which incorporates classifier-free guidance:

$$\frac{dz_t}{dt} = f(t)\, z_t + \frac{g^2(t)}{2\sigma_t}\, \tilde{\epsilon}_\theta(z_t, \omega, c, t), \quad z_T \sim \mathcal{N}(0, \tilde{\sigma}^2 \mathbf{I}), \tag{3}$$

where $\tilde{\epsilon}_\theta(z_t, \omega, c, t) = (1+\omega)\, \epsilon_\theta(z_t, c, t) - \omega\, \epsilon_\theta(z_t, \varnothing, t)$ is the guided noise prediction and ω is the guidance scale. LCMs are trained by minimizing the latent consistency distillation loss:

$$\mathcal{L}_{\text{LCD}} = \mathbb{E}_{z,c,\omega,n}\Big[d\Big(f_\theta(z_{t_{n+k}}, \omega, c, t_{n+k}),\ f_{\theta^-}(\hat{z}_{t_n}^{\Psi,\omega}, \omega, c, t_n) \Big) \Big] \tag{4}$$

where $\hat{z}_{t_n}^{\Psi,\omega}$ is estimated from $z_{t_{n+k}}$ using an ODE solver Ψ (e.g., DDIM), θ^- denotes an exponential moving average of the parameters, $d(\cdot,\cdot)$ is a distance metric, and k is a skipping-step interval that accelerates convergence by enforcing consistency over larger timestep gaps.

A key property of LCMs for our work is that they distill the multi-step PF-ODE solving process into as few as 2 to 4 forward passes while faithfully approximating the solution of the pretrained model's PF-ODE. This compression is the key property we exploit: gradients can flow from a downstream loss through the few LCM steps back to the input latent, which is computationally prohibitive with N-step DDIM sampling.

3. DiffMark: Differentiable Watermarking

Given an L-bit secret $s \in \{0,1\}^L$, our goal is to embed s into a diffusion-generated images satisfying three requirements: (i) s is recoverable in a **single-forward pass**; (ii) the watermarked image is **visually indistinguishable** from a clean sample, and (iii) the watermark is embedded **without modifying** the weights of the underlying DM. DiffMark achieves these goals through two core ideas. First, instead of encoding the watermark into the initial noise z_T (Wen et al.,

2023; Li et al., 2025; Ci et al., 2024), which necessitates costly inversion for recovery, we embed it as a persistent learned perturbation δ at every denoising step (Sec. 3.1). This perturbation accumulates in the final denoised latent z_0 and can be extracted by a single forward pass through a lightweight decoder D_ψ. Second, to enable end-to-end encoder training through the frozen UNet, we propose to employ LCM as a differentiable training bridge (Sec. 3.2). Additionally, we introduce a multi-stage curriculum training strategy to prevent the training collapse (Sec. 3.3). Fig. 1 provides an overview of DiffMark at inference time and its dual-path training strategy.

3.1. Watermark Embedding via Persistent Delta Injection

3.1.1. Delta Injection Mechanism

Given an L-bit secret $s \in \{0,1\}^L$, a lightweight encoder E_ϕ maps s to a latent-space perturbation $\delta = E_\phi(s) \in \mathbb{R}^{c\times h\times w}$, where c, h, w match the dimensions of the diffusion latent space. The encoder is called once, and the same δ is reused at every denoising step. We modify the standard DDIM sampling process (Song et al., 2020) as follows. Let $\{t_1, t_2, \ldots, t_N\}$ denote the denoising timestep schedule- and let $z_T \sim \mathcal{N}(0, \mathbf{I})$ be the initial noise. At each step $k = 1, \ldots, N$, the watermarked denoising trajectory is:

$$\tilde{z}_{t_k} = z_{t_k} + \delta, \tag{5}$$

$$\hat{\epsilon}_{t_k} = (1+w)\,\epsilon_\theta(\tilde{z}_{t_k}, t_k, c) - w\,\epsilon_\theta(\tilde{z}_{t_k}, t_k, \varnothing), \tag{6}$$

$$z_{t_{k+1}} = \sqrt{\bar{\alpha}_{t_{k+1}}}\left(\frac{\tilde{z}_{t_k} - \sqrt{1-\bar{\alpha}_{t_k}}\,\hat{\epsilon}_{t_k}}{\sqrt{\bar{\alpha}_{t_k}}}\right) \tag{7}$$

$$+ \sqrt{1-\bar{\alpha}_{t_{k+1}}}\,\hat{\epsilon}_{t_k}, \tag{8}$$

where ϵ_θ is the frozen UNet, c is the text conditioning, $\varnothing$ denotes the null condition for classifier-free guidance with scale w, and $\bar{\alpha}_t$ is the cumulative noise schedule. Because δ is injected *before* each UNet evaluation, the noise prediction $\hat{\epsilon}_{t_k}$ is conditioned on the perturbed latent $\tilde{z}_{t_k}$, and its effect on the denoising trajectory accumulates through subsequent DDIM updates (Eq. (8)), progressively shaping the final latent z_0. A lightweight decoder D_ψ then recovers the secret in a single forward pass: $\hat{s} = D_\psi(z_0) \in \mathbb{R}^{L\times 2}$, enabling multi-bit capacity for user identification without costly inversion.

3.1.2. Perturbation Regularization

Unlike sampling-based methods that replace z_T with a watermark-carrying noise vector, DiffMark leaves $z_T \sim \mathcal{N}(0, \mathbf{I})$ entirely unmodified. To keep the UNet operating within its trained regime and minimize perceptible artifacts in the generated image, we regularize δ through two complementary constraints. First, a magnitude loss penalizes deviations of δ's standard deviation from a target value σ_{target}:

$$\mathcal{L}_{\text{mag}} = \big(\sigma(\delta) - \sigma_{\text{target}}\big)^2, \tag{9}$$

where σ_{target} is annealed from a relaxed initial value to a tighter final value over training (Appendix D.1). Second, a KL divergence term regularizes the encoder's variational distribution toward the standard Gaussian:

$$\begin{aligned}\mathcal{L}_{\text{KL}} &= \text{KL}\big(q_\phi(\delta \mid s) \,\|\, \mathcal{N}(0, \mathbf{I})\big) \\ &= -\frac{1}{2|\delta|}\sum_{j=1}^{|\delta|}\Big(1 + \log\sigma_j^2 - \mu_j^2 - \sigma_j^2\Big),\end{aligned} \tag{10}$$

where μ_j and σ_j^2 are the mean and variance produced by the encoder's variational heads. Together, these terms enforce $\|\delta\| \ll \|z_T\|$, ensuring that each perturbed input $\tilde{z}_{t_k} = z_{t_k} + \delta$ remains close to the clean trajectory.

3.1.3. Encoder-Decoder Pretraining

Prior to full training with the diffusion model, the encoder and decoder are jointly pretrained to establish a reliable secret-to-perturbation mapping. The encoder produces $\delta = E_\phi(s)$ for a randomly sampled secret $s \sim \text{Bernoulli}(0.5)^L$, and the decoder is trained to recover s from both δ and noisy variants $\delta + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma_n^2\mathbf{I})$, where σ_n increases over training. The decoder is supervised with the per-bit cross-entropy loss:

$$\mathcal{L}_{\text{CE}}(\mathbf{o}, s) = -\frac{1}{L}\sum_{i=1}^{L}\sum_{c\in\{0,1\}} \mathbb{1}[s_i = c]\,\log p_{i,c}, \tag{11}$$

where $\mathbf{o} = D_\psi(\cdot) \in \mathbb{R}^{L\times 2}$ and $p_{i,c} = \frac{\exp(\mathbf{o}_{i,c})}{\exp(\mathbf{o}_{i,0})+\exp(\mathbf{o}_{i,1})}$. We further introduce an orthogonality loss $\mathcal{L}_{\text{orth}}$ to prevent the encoder from mapping all secrets to the same perturbation. This loss is defined as the mean pairwise cosine similarity between perturbations within a batch:

$$\mathcal{L}_{\text{orth}} = \frac{1}{B(B-1)}\sum_{i\neq j}\frac{\langle \delta_i, \delta_j\rangle_F}{\|\delta_i\|_F\|\delta_j\|_F}, \tag{12}$$

where $\langle\cdot,\cdot\rangle_F$ and $\|\cdot\|_F$ denote the Frobenius inner product and norm, respectively. Minimizing $\mathcal{L}_{\text{orth}}$ encourages different secrets to produce orthogonal perturbations, ensuring that the decoder can distinguish them. Details in encoder-decoder architectures are provided in Appendix C.

3.2. LCM as a Differentiable Training Bridge

The delta injection mechanism described in Sec. 3.1 requires backpropagating gradients from the decoder D_ψ through the entire denoising trajectory to the encoder E_ϕ. Standard DDIM sampling with $N = 50$ steps creates a computational graph of N sequential UNet evaluations, making this prohibitive in both memory and gradient stability. Our key

observation is that the encoder does not require the full N-step DDIM path for gradient computation; it only needs a ***differentiable approximation*** that faithfully represents how δ influences z_0.

We address this by employing Latent Consistency Models (Luo et al., 2023) as a differentiable training bridge. However, LCM's few-step approximation produces latents of lower fidelity than full DDIM. We therefore introduce a **dual-path training strategy** that decouples encoder optimization from decoder calibration: a short, differentiable LCM path provides the encoder with a tractable gradient signal, while a parallel full-step DDIM path supplies the decoder with high-fidelity supervision.

3.2.1. LCM PATH (DIFFERENTIABLE)

Starting from $z_{t_1} = z_T$, each of the $K = 4$ LCM steps applies delta injection followed by the frozen LCM forward:

$$\begin{aligned} \tilde{z}_{t_k} &= z_{t_k} + \delta, \\ z_{t_{k+1}} &= \mathrm{LCM}_\theta(\tilde{z}_{t_k},\, t_k,\, c), \quad k = 1, \ldots, K, \end{aligned} \tag{13}$$

yielding the denoised latent z_0^{lcm}. Because the LCM forward pass is fully differentiable, the reconstruction loss $\mathcal{L}_{\mathrm{lcm}} = \mathcal{L}_{\mathrm{CE}}(D_\psi(z_0^{\mathrm{lcm}}), s)$ provides primary gradient signal to E_ϕ via the entire chain:

$$\begin{aligned} \nabla_\phi \mathcal{L}_{\mathrm{lcm}} : \mathcal{L}_{\mathrm{lcm}} &\xrightarrow{\nabla} D_\psi \xrightarrow{\nabla} z_0^{\mathrm{lcm}} \\ &\xrightarrow{\nabla} \underbrace{\text{LCM step } K \;\rightarrow\; \cdots \;\rightarrow\; \text{LCM step } 1}_{K \text{ differentiable steps}} \\ &\xrightarrow{\nabla} \delta \xrightarrow{\nabla} E_\phi \end{aligned} \tag{14}$$

The frozen UNet weights θ are never updated; gradients pass *through* the UNet but do not modify it.

3.2.2. DDIM PATH (NON-DIFFERENTIABLE).

In parallel, the standard N-step DDIM sampler (N=50) runs with the same z_T and a stop-gradient copy $\bar{\delta} = \mathrm{sg}(\delta)$, which each injection scaled by $1/N$ to match the cumulative perturbation of the LCM path:

$$\begin{aligned} \tilde{z}_{t_k} &= z_{t_k} + \tfrac{\bar{\delta}}{N}, \\ z_{t_{k+1}} &= \mathrm{DDIM}_\theta(\tilde{z}_{t_k},\, t_k,\, c), \quad k = 1, \ldots, N. \end{aligned} \tag{15}$$

The resulting high-fidelity latent z_0^{ddim} drives the DDIM supervision loss $\mathcal{L}_{\mathrm{ddim}} = \mathcal{L}_{\mathrm{CE}}\big(D_\psi(z_0^{\mathrm{ddim}}),\, s\big)$, which trains the decoder on realistic full-quality outputs without propagating gradients to the encoder.

Together, the LCM path teaches the encoder *where* to place the watermark signal via a short, differentiable computational graph, while the DDIM path teaches the decoder *how* to extract it from the high-fidelity latents it will encounter at inference.

3.2.3. IMPERCEPTIBILITY PRESERVATION

Because δ is injected in the latent space of the DM, even small deviations in z_0 can be amplified by the nonlinear VAE decoder into perceptible pixel-space artifacts. We design a *latent fidelity* loss to penalize distortion directly in latent space, before the lossy decoding step:

$$\mathcal{L}_{\mathrm{lafid}} = \mathrm{MSE}(z_0^{\mathrm{lcm}},\, z_0^{\mathrm{lcm,clean}}), \tag{16}$$

where $z_0^{\mathrm{lcm,clean}}$ is the LCM output from the same z_T with zero perturbation. While $\mathcal{L}_{\mathrm{lafid}}$ controls global latent distortion, the watermark energy may still concentrate in localised pixel regions after VAE decoding. We therefore additionally employ the peak regional variational loss (Feng et al., 2024) to distribute the watermark energy across the entire image. Finally, inspired by the work (Li et al., 2025), we also constrain the frequency-domain characteristics of δ into high frequencies, where the human visual system is least sensitive. Details about these loss functions can be found in Appendix E.

3.2.4. FALSE-POSITIVE SUPPRESSION

To prevent the decoder from outputting confident predictions on *any* input, we apply a *negative entropy* loss on non-watermarked latents z_0^{clean} produced by DDIM without delta injection:

$$\mathcal{L}_{\mathrm{neg}} = -\frac{1}{BL} \sum_{b,i} H\big(D_\psi(z_0^{\mathrm{clean}})_{b,i}\big), \tag{17}$$

where $H(\cdot)$ denotes the per-bit entropy of the softmax output. Maximising the decoder's output entropy on clean images drives the per-bit predictions toward a uniform distribution over $\{0, 1\}$, ensuring that unwatermarked content yields near-chance bit accuracy and thus prevents false detection.

3.3. Multi-stage Curriculum Training Strategy

Training DiffMark requires jointly optimizing two competing objectives: watermark detection accuracy and imperceptibility. Naively activating all objectives from initialization leads to training collapse: imperceptibility losses drive $\|\delta\| \to 0$, whereas the reconstruction losses require $\|\delta\|$ to remain sufficiently large for reliable decoding. We resolve this conflict by partitioning the objectives into two curriculum-gated groups,

$$\begin{aligned} \mathcal{G}_{\mathrm{rec}} &= \{\mathcal{L}_{\mathrm{lcm}},\, \mathcal{L}_{\mathrm{ddim}}\}, \\ \mathcal{G}_{\mathrm{imp}} &= \{\mathcal{L}_{\mathrm{lafid}},\, \mathcal{L}_{\mathrm{prvl}},\, \mathcal{L}_{\mathrm{freq}},\, \mathcal{L}_{\mathrm{neg}}\}, \end{aligned} \tag{18}$$

and activating them in a strict order. Each loss $\mathcal{L}_i$ is gated by $g_i(t) = \mathbb{1}[t \geq \tau_i]$, where τ_i is its activation step. The schedule satisfies: $\max_{i \in \mathcal{G}_{\mathrm{rec}}} \tau_i \leq \min_{j \in \mathcal{G}_{\mathrm{imp}}} \tau_j$, so the encoder–decoder pair first establishes a decodable watermark, then refines it for imperceptibility. The total loss at step t is: $\mathcal{L}(t) = \sum_i g_i(t) \cdot w_i(t) \cdot \mathcal{L}_i$, where $w_i(t)$ is the weight for each term.

Table 1. Quantitative comparison on DiffusionDB. "Plug&Play" indicates whether the method operates on a frozen, unmodified diffusion model. Detection accuracy is measured on clean (unattacked) watermarked images. Generation consistency metrics compare watermarked vs. unwatermarked images from the same noise. **Best** results among plug-and-play methods are **bolded**; overall best are underlined.

Method	Type	Plug &Play	Multi-bit	Detection Accuracy			Generation Consistency		Generation Quality	
				Bit Acc (Clean)	TPR@1%FPR (Clean)	TPR@0.1%FPR (Clean)	PSNR↑	LPIPS↓	FID↓	CLIP-FID↓
StegaStamp	Post Generation	✗	100 bits	0.9994	1.0	1.0	11.34	0.7171	54.82	10.26
Stable Signature	Fine-tuning	✗	48 bits	0.9950	0.9999	0.9900	16.23	0.5164	46.81	4.61
AquaLoRA		✗	48 bits	0.9355	0.9970	0.9910	20.59	0.4791	32.32	1.98
Tree-Ring	Sampling	✓	0 bit	—	1.0	1.0	11.02	0.7441	47.09	4.65
RingID		✓	11 bits	—	1.0	1.0	10.74	0.7481	47.18	4.77
Shallow Diffuse		✓	0 bit	—	1.0	1.0	**11.01**	0.7469	43.37	4.10
DiffMark (Ours)		✓	**64 bits**	**0.9381**	**1.0**	**1.0**	**11.01**	**0.7224**	**38.07**	**2.20**

4. Experiments

4.1. Experimental Settings

Baselines & Datasets. We utilize 3 public datasets for evaluation: MS-COCO 2017 (Lin et al., 2014), DiffusionDB (Wang et al., 2023), and DALL-E3 (An et al., 2024). For each dataset, we select 1000 images for evaluation. 10,000 images from DiffusionDB are utilized for training. Regarding baselines, we compare with 5 methods, including StegaStamp (Tancik et al., 2020), Stable Signature (Fernandez et al., 2023), AquaLoRA robust version (Feng et al., 2024), Tree-Ring (Wen et al., 2023), RingID (Ci et al., 2024), and Shallow Diffuse (Li et al., 2025). We mainly report results on DiffusionDB in the main paper and leave results on other datasets in Appendix G.

Evaluation Metrics. To evaluate watermark detection accuracy, we calculate the bit accuracy (Bit ACC) and TPR@0.1%FPR. For image quality evaluation, we use PSNR (Jähne, 2005), LPIPS (Zhang et al., 2018), FID (Heusel et al., 2017) and CLIP-FID (An et al., 2024).

Implementation Details. We adopt Stable Diffusion v1.5 (Rombach et al., 2022) as the base diffusion model with its components kept frozen throughout training and inference. For the differentiable LCM bridge, we use LCM_Dreamshaper_v7 (Luo et al., 2023) with K=4 denoising steps. During pretraining, the encoder and decoder are jointly trained for up to 50,000 (batch size 64) steps using AdamW at 3×10^{-4} (encoder) and 1×10^{-4} (decoder). Fine-tuning with DM runs for 10,000 steps with batch size 16. We use AdamW with learning rates 5×10^{-5} (encoder) and 3×10^{-4} (decoder), linear warmup over 500 steps followed by linear decay to 10^{-6}. More details are provided in Appendix F.

4.2. Watermark Detection and Image Quality

Tab. 1 compares all methods on clean watermarked images generated from DiffusionDB prompts. Compared to sampling-based methods, DiffMark achieves a perfect TPR of 1.0 at both 1% and 0.1% FPR thresholds and better generation quality (e.g., CLIP-FID 2.20) while additionally providing 64-bit multi-bit capacity and single-pass detection (Sec. 4.5). Notably, DiffMark achieves comparable per-bit accuracy to AquaLoRA (0.9381 vs. 0.9355) while embedding a strictly longer secret (64 bits vs. 48 bits), yielding a larger identification key space (2^{64} vs. 2^{48}) without sacrificing detection reliability. Although fine-tuning-based methods (i.e., Stable Signature and AquaLoRA) achieve competitive detection accuracy and the best generation consistency, they are tied to a single DM and cannot transfer across architectures without retraining/fine-tuning. DiffMark overcomes this limitation, as demonstrated by the cross-model analysis in Sec. 4.4.

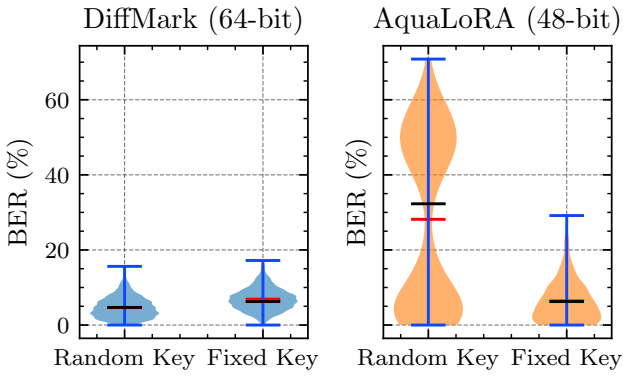


Figure 2. Bit Error Rate (BER) distributions for fixed-key and random-key generation across methods. Each violin shows the per-image BER over 1,000 images; wider regions indicate higher density.

4.3. Key Flexibility

In this experiment, we show that DiffMark can decode an ***arbitrary*** secret embedded at generation time, not merely a single predetermined key used during training. For each method, we generate two sets of 1,000 images from DiffusionDB prompts: a *fixed-key* set, in which every image carries the same predetermined secret, and a *random-key* set, in which each image is assigned an independent uniformly random secret. From Fig. 2, we can observe that DiffMark's performance is consistent regardless of whether

a predetermined or arbitrary per-image key is used. Furthermore, per-image BER is uncorrelated with the Hamming distance between the runtime secret and the training key, demonstrating genuine generalisation across the full 2^{64} key space rather than interpolation near a fixed anchor (Appendix G.4). In contrast, AquaLoRA degrades sharply from 6.42% (fixed) to 28.16% (random), exposing overfitting to their training key.

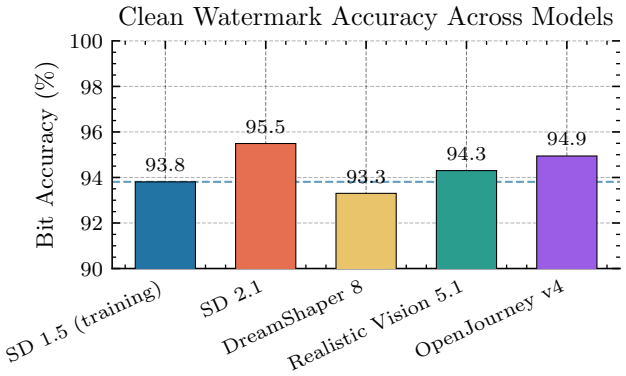


Figure 3. Cross-model transferability of DiffMark. Bit accuracy on four unseen SD-family models after training exclusively on SD 1.5, with no per-model fine-tuning.

4.4. Cross-Model Transferability

This section demonstrates that DiffMark can generalize beyond the model on which it was trained. In contrast to fine-tuning-based methods such as Stable Signature, which requires retraining the UNet decoder for each target model, and AquaLoRA, which requires fitting model-specific LoRA modules, DiffMark imposes no such overhead. We evaluate this property by training exclusively on SD 1.5 and testing on four unseen models, including SD-2.1 (Rombach et al., 2022), DreamShaper 8 (Lykon, 2023), Realistic Vision 5.1 (Vision, 2023), and OpenJourney v4 (PromptHero, 2023), without any fine-tuning. As shown in Fig. 3, DiffMark achieves 93.3–95.5% bit accuracy across all target models, confirming that DiffMark is a genuinely plug-and-play solution deployable across various SD-family models without per-model retraining. The evaluation under the attack context is provided in Appendix.

4.5. Watermark Detection Latency

Fig. 4 reports per-image decode latency measured on an L40S GPU averaged over 100 images. Sampling-based methods incur latencies of 754.9 ms (Tree-Ring), 753.2 ms (RingID), and 239.8 ms (Shallow Diffuse), as all three require running N-step DDIM inversion to recover the initial noise vector before pattern matching. DiffMark reduces detection latency to 16.4 ms, achieving a $45\times$ speedup over these methods. Compared to fine-tuning-based methods that also achieve single-pass detection, DiffMark incurs a modest overhead attributable to the additional VAE encoding step ($z_0 = \mathcal{E}(x) \cdot f_s$) absent in methods that embed the watermark directly in pixel space.

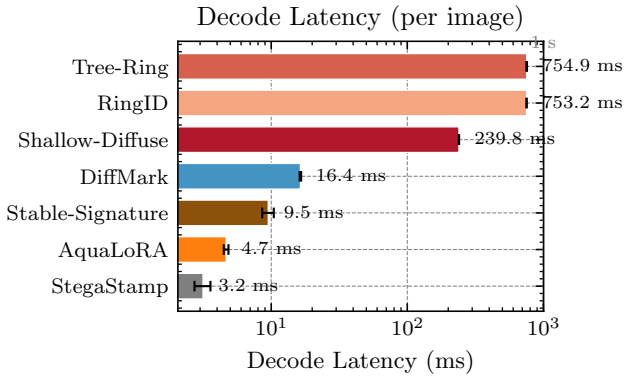


Figure 4. Decode latency comparison (per image, L40S GPU). Latencies are averaged over 100 images; the x-axis is on a log scale.

4.6. Identification Analysis

In a deployment scenario with N registered users, each user $i \in \{1, \cdots, N\}$ is assigned a unique L-bit secret key $s^{(i)} \in \{0,1\}^L$, and the set of all registered keys is denoted $\mathcal{S} = \{s^{(1)}, \cdots, s^{(N)}\}$. Given a query image x, the decoder produces a soft prediction $\hat{s} = \arg\max D_\psi(\mathcal{E}(x))$, where $\mathcal{E}$ denotes the VAE encoder. The identification task is to determine which user generated the image, i.e., to find $i^* = \arg\min_i d_H(\hat{s}, s^{(i)})$, where $d_H(\cdot,\cdot)$ denotes Hamming distance. Identification succeeds when the decoded secret $\hat{s}$ is closer to the true key $s^{(i)}$ than to every other registered key. Detailed experimental design is provided in Appendix G.3.1. Fig. 5 evaluates the clean (no-attack) setting, scaling the number of registered users from N=10 to N=10^8. DiffMark achieves perfect Top-1 accuracy up to 10^6 users and maintains $\geq$99.97% at 10^8 users, confirming that the 64-bit secret provides ample identification capacity for platform-scale deployment.

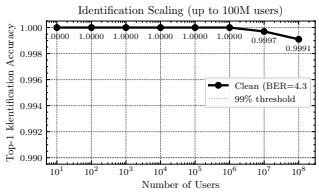


Figure 5. Clean identification scaling. Top-1 identification accuracy as a function of database size (number of registered users) without any attack.

4.7. Robustness

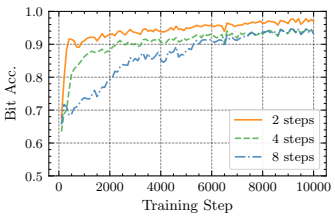
Tab. 2 shows that DiffMark also achieves near-perfect robustness against the majority of the 13 attack types evaluated, despite robustness not being its primary design objective. DiffMark attains perfect TPR on all photometric distortion attacks (brightness, contrast, JPEG, Gaussian noise, random erasing) and on the two strongest regeneration attacks (Regen-Diff, Rinse-2×Diff). These results confirm

Table 2. Robustness evaluation across 13 attacks on DiffusionDB (TPR@0.1%FPR).

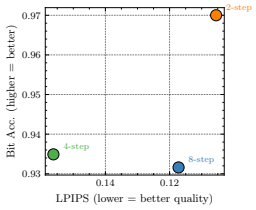
Attack	Type	StegaStamp	Stable Signature	AquaLoRA	Tree-Ring	RingID	Shallow Diffuse	DiffMark (Ours)
Bright	Distortion Attacks	1.00	1.00	0.96	0.74	1.00	1.00	**1.00**
Compress		**1.00**	**1.00**	0.99	0.74	1.00	1.00	**1.00**
Contrast		**1.00**	**1.00**	0.97	0.74	0.98	1.00	**1.00**
Erase		**1.00**	**1.00**	0.96	0.51	**1.00**	**1.00**	**1.00**
RCrop		0.39	**1.00**	0.91	0.03	0.01	0.00	0.00
Rotation		0.00	0.65	0.00	0.16	**1.00**	0.00	0.00
Blur		0.48	0.00	0.96	0.23	**1.00**	0.00	0.00
Noise		**1.00**	0.99	0.96	0.79	0.99	**1.00**	0.99
Regen-VAE	Regeneration Attacks	**1.00**	0.36	0.94	0.51	0.97	0.87	0.81
Regen-Diff		0.23	0.02	0.78	0.80	0.94	1.00	**1.00**
Rinse-2Xdiff		0.11	0.01	0.52	0.82	0.81	0.99	**1.00**
Adv-KLVAE8	Adversarial Attacks	0.26	**1.00**	0.98	0.29	0.31	0.60	0.31
Adv-RN18		0.17	0.98	0.99	0.87	0.37	0.58	**1.00**
Average		0.59	0.69	**0.84**	0.56	0.80	0.70	0.70

the advantages of embedding into the latent space rather than pixel one (Feng et al., 2024). With an average TPR of 0.70, DiffMark outperforms Tree-Ring (0.56) and StegaStamp (0.59) and remains comparable to RingID (0.80) and Stable Signature (0.69), while uniquely offering per-image key flexibility and cross-model transferability without any fine-tuning, capabilities that none of these baselines provide simultaneously.

The three attacks where DiffMark scores zero (rotation, blur, resized crop) share a common characteristic: they introduce large geometric or spatial-frequency distortions that corrupt the latent encoding $z_0 = \mathcal{E}(x) \cdot f_s$ before the decoder can operate. This limitation also affects Shallow Diffuse. Since the grey-box adversarial attack Adv-KLVAE8 directly targets the VAE encoder, it disrupts the latent representations on which DiffMark's decoder depends.



(a) Training curves for $K \in \{2, 4, 8\}$. LCM steps.



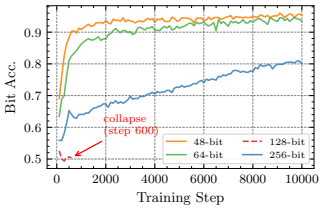
(b) Image Quality vs. Detection Accuracy Trade-off.

Figure 6. LCM Step Ablation

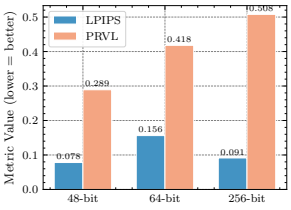
4.8. Ablation Study

4.8.1. LCM Step Ablation

Fig. 6 examines how the number of LCM bridge steps K affects the accuracy–imperceptibility trade-off. The main experiments in Secs. 4.2–4.7 use $K = 4$, which was the conservative setting adopted in our initial experimental configuration: it provides a short differentiable path while retaining a closer few-step approximation to the full denoising trajectory than $K = 2$. However, the ablation reveals that $K = 2$ is a particularly attractive operating point. It converges faster, reduces memory and training cost, and achieves competitive detection accuracy. Since the full DDIM supervision and inference paths remain fixed at $N = 50$, the role of the LCM path is primarily to provide a stable encoder-gradient bridge rather than to serve as the final high-fidelity sampler. From this perspective, increasing K beyond two steps can lengthen the gradient path without necessarily improving final DDIM-latent decoding.



(a) Training curves for $L \in \{48, 64, 128, 256\}$ bit.



(b) Quality–capacity tradeoff across bit length.

Figure 7. Bit Length Ablation.

We therefore keep $K = 4$ for the reported main comparisons to maintain consistency with the experimental setting used throughout this version, while identifying $K = 2$ as a promising default for the extended version, where all hyperparameters and main experiments can be re-tuned around the shorter bridge.

4.8.2. Bit Length Ablation

Fig. 7 reveals that $L = 128$ suffers a training collapse at step 600, where $\mathcal{L}_{\text{orth}}$ can no longer maintain diverse perturbations within the latent budget, driving $\|\delta\| \to 0$. The image quality gap between 48 and 64 bits is modest, while 256 bits incurs a severe quality penalty (LPIPS 0.508). We adopt $L = 64$ as it provides sufficient capacity for identification across 2^{64} keys while preserving competitive image quality.

5. AI Governance Implications

Generative-AI governance is increasingly moving from broad transparency principles to operational requirements for provenance, monitoring, and accountability. Recent frameworks require synthetic content to be marked or detectable, provenance records to be verifiable, and providers to support post-deployment audit and incident response (European Parliament and Council of the European Union, 2024b; Coalition for Content Provenance and Authenticity, 2024). However, these goals are difficult to realize without technical mechanisms that remain usable after content dissemination, support many users, and can be verified at the platform scale. Our results suggest that DiffMark can serve as such a mechanism for diffusion-based image generation.

First, DiffMark turns watermarking from a binary label into an attribution primitive. Zero-bit watermarks can support the question "was this image AI-generated?", but they cannot identify the relevant accountable principal when many users, accounts, model endpoints, or fine-tuned derivatives share the same generation infrastructure. By embedding a runtime 64-bit secret, DiffMark allows each generated image to carry a pseudonymous key that can be mapped, in a secure registry, to a user account, deployment, model endpoint, or policy context. This directly supports governance workflows for traceability and incident response: when harmful synthetic content is discovered, an authorized auditor can recover the key and rank candidate principals by Hamming distance, rather than merely infer that the image was generated by AI.

Second, DiffMark makes attribution practical at scale. Governance mechanisms fail if verification is too costly for routine deployment. Sampling-based methods require tens of DDIM inversion steps per image, which is expensive for large content streams. In contrast, DiffMark decodes the embedded secret in a single latent-space forward pass, reducing detection latency to 16.4 ms per image (Sec. 4.5). This enables batch auditing by platforms, post-hoc investigation by regulators or trusted third parties, and large-scale monitoring without re-running the generator or accessing prompts, seeds, or model weights. The identification analysis in Sec. 4.6 further shows that the 64-bit key space supports platform-scale attribution, maintaining high Top-1 identification accuracy even as the registered population grows to very large user sets.

Third, the frozen-model and cross-model nature of DiffMark supports standardized oversight in a fragmented model ecosystem. Fine-tuning-based watermarks are tied to specific checkpoints and therefore require re-watermarking or retraining whenever a model is updated, customized, or released as an open-source derivative. This is a poor fit for governance, where oversight tools should ideally remain stable across rapidly changing model families. Because DiffMark leaves the underlying diffusion model unmodified and transfers across unseen SD-family models without per-model fine-tuning (Sec. 4.4), a single encoder–decoder pair can support a more uniform watermarking and detection protocol across multiple deployments. This reduces compliance costs for developers and enables more consistent verification for auditors, platforms, and standards bodies.

Fourth, DiffMark complements, rather than replaces, cryptographic provenance mechanisms such as C2PA. Metadata-based provenance is well-suited for signed, user-facing records of origin and editing history, while latent watermarking provides a content-bound signal that can be recovered when metadata is unavailable to the verifier. A practical governance stack can therefore use both layers: C2PA-style credentials for transparent disclosure and DiffMark as a forensic fallback for images that have been rehosted, recompressed, or separated from their original distribution context. This layered design better matches real-world media circulation, where no single provenance mechanism is sufficient under all threat models.

Finally, watermark-based governance should be deployed with appropriate safeguards. DiffMark keys should be pseudonymous, with identity mapping performed only by authorized parties through an access-controlled and auditable registry. Detection decisions should be calibrated at fixed false-positive rates and should not be treated as the sole basis for enforcement, especially under attacks that corrupt the VAE latent representation, such as large rotations, blur, or crop-and-resize transformations (Sec. 4.7). These limitations do not undermine the governance value of the method; rather, they clarify its role as one layer in a broader evidence pipeline combining cryptographic credentials, platform logs, human review, and legal process.

6. Conclusion

We present DiffMark, a plug-and-play multi-bit watermarking method that bridges the gap between emerging AI governance mandates and practical technical tooling. By injecting a persistent learned perturbation at every denoising step of a frozen UNet, DiffMark accumulates a robust signal without modifying model weights. Crucially, DiffMark provides single-pass 64-bit detection at 16.4 ms, per-image key flexibility and cross-model transferability without fine-tuning, directly enabling scalable user attribution and standardized oversight across a fragmented open-source AI ecosystem.

Acknowledgements

This work was funded by Taighde Eireann – Research Ireland through the Research Ireland Centre for Research Training in Machine Learning (18/CRT/6183).

References

AI Seoul Summit. Seoul declaration: International network of AI safety institutes, 2024.

Al-Haj, A. Combined dwt-dct digital image watermarking. *Journal of computer science*, 3(9):740–746, 2007.

An, B., Ding, M., Rabbani, T., Agrawal, A., Xu, Y., Deng, C., Zhu, S., Mohamed, A., Wen, Y., Goldstein, T., et al. Waves: Benchmarking the robustness of image watermarks. In *Forty-first International Conference on Machine Learning*, 2024.

Bui, T., Agarwal, S., and Collomosse, J. Trustmark: Universal watermarking for arbitrary resolution images. *arXiv preprint arXiv:2311.18297*, 2023.

California State Legislature. Senate bill no. 53: Transparency in frontier artificial intelligence act (TFAIA), 2025. Chapter 138, Statutes of 2025.

China AI Safety and Development Association. China AI safety and development association (CnAISDA), 2025.

Ci, H., Yang, P., Song, Y., and Shou, M. Z. Ringid: Rethinking tree-ring watermarking for enhanced multi-key identification. In *European conference on computer vision*, pp. 338–354. Springer, 2024.

CNN. Finance worker pays out $25 million after video call with deepfake CFO. `https://edition.cnn.com/2024/02/04/asia/deepfake-cfo-scam-hong-kong-intl-hnk/index.html`, 2024. Accessed: 28 June 2025.

Coalition for Content Provenance and Authenticity. C2PA technical specification, version 2.1. `https://c2pa.org/specifications/specifications/2.1/specs/C2PA_Specification.html`, 2024.

Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

European Parliament and Council of the European Union. Regulation (EU) 2024/1689 – the artificial intelligence act, 2024a. Official Journal of the European Union.

European Parliament and Council of the European Union. Regulation (EU) 2024/1689 of the European Parliament and of the Council – the artificial intelligence act, 2024b. Official Journal of the European Union, Article 50.

Feng, W., Zhou, W., He, J., Zhang, J., Wei, T., Li, G., Zhang, T., Zhang, W., and Yu, N. Aqualora: toward white-box protection for customized stable diffusion models via watermark lora. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 13423–13444, 2024.

Fernandez, P., Sablayrolles, A., Furon, T., Jégou, H., and Douze, M. Watermarking images in self-supervised latent spaces. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3054–3058. IEEE, 2022.

Fernandez, P., Couairon, G., Jégou, H., Douze, M., and Furon, T. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22466–22477, 2023.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Huang, H., Wu, Y., and Wang, Q. Robin: Robust and invisible watermarks for diffusion models with adversarial optimization. *Advances in Neural Information Processing Systems*, 37:3937–3963, 2024.

Jähne, B. *Digital image processing*. Springer, 2005.

Jia, Z., Fang, H., and Zhang, W. Mbrs: Enhancing robustness of dnn-based watermarking by mini-batch of real and simulated jpeg compression. In *Proceedings of the 29th ACM international conference on multimedia*, pp. 41–49, 2021.

Li, W., Zhang, H., and Qu, Q. Shallow diffuse: Robust and invisible watermarking through low-dim subspaces in diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Luo, S., Tan, Y., Huang, L., Li, J., and Zhao, H. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Lykon. DreamShaper 8. https://huggingface.co/Lykon/dreamshaper-8, 2023.

National Technical Committee 260 on Cybersecurity of Standardization Administration of China (SAC/TC260). Basic security requirements for generative artificial intelligence service, 2024.

Navas, K., Ajay, M. C., Lekshmi, M., Archana, T. S., and Sasikumar, M. Dwt-dct-svd based watermarking. In *2008 3rd international conference on communication systems software and middleware and workshops (COMSWARE'08)*, pp. 271–274. IEEE, 2008.

Neekhara, P., Hussain, S., Zhang, X., Huang, K., McAuley, J., and Koushanfar, F. Facesigns: semi-fragile neural watermarks for media authentication and countering deepfakes. *arXiv preprint arXiv:2204.01960*, 2022.

New York State Legislature. Responsible AI safety and education (RAISE) act (s6953b/a6453b), 2025.

Nguyen-Le, H.-H., Tran, V.-T., Nguyen, T., and Le-Khac, N.-A. A survey on proactive deepfake defense: Disruption and watermarking. *ACM Computing Surveys*, 58(5): 1–37, 2025.

Pearson, J. and Zinets, N. Deepfake footage purports to show ukrainian president capitulating. https://www.reuters.com/world/europe/deepfake-footage-purports-show-ukrainian-president-capitulating-2022-03-16, 2022. Accessed: 12 February 2026.

Pezenik, S. and Shepherd, B. Fake biden robocall urges new hampshire voters to skip their primary. https://abcnews.com/Politics/fake-biden-robocall-urges-new-hampshire-voters-skip/story?id=106580926, 2024. Accessed: 12 February 2026.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Sdxl: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

PromptHero. OpenJourney v4. https://huggingface.co/prompthero/openjourney-v4, 2023.

Reuel, A., Soder, L., Bucknall, B., and Undheim, T. A. Position: Technical research and talent is needed for effective ai governance. In *Forty-first International Conference on Machine Learning*, 2024.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22500–22510, 2023.

Sander, T., Fernandez, P., Durmus, A., Furon, T., and Douze, M. Watermark anything with localized messages. In *International Conference on Learning Representations-ICLR 2025*, 2025.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

Song, Y., Dhariwal, P., Chen, M., and Sutskever, I. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 32211–32252, 2023.

Tancik, M., Mildenhall, B., and Ng, R. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2117–2126, 2020.

Tenbarge, K. Explicit, AI-generated taylor swift images spread rapidly on social media. https://www.nbcnews.com/tech/tech-news/explicit-ai-generated-taylor-swift-images-continue-proliferate-x-insta-rcna136193, 2024. Accessed: 28 June 2025.

UK Department for Science, Innovation and Technology. Introducing the AI safety institute, 2023.

U.S. National Institute of Standards and Technology (NIST). U.S. artificial intelligence safety institute, 2023.

Vision, R. Realistic Vision V5.1. https://huggingface.co/SG161222/Realistic_Vision_V5.1_noVAE, 2023.

Wang, T., Huang, M., Cheng, H., Zhang, X., and Shen, Z. Proactive deepfake detection via training-free landmark perceptual watermarks. In *ACM Multimedia 2024*, 2024.

Wang, Z. J., Montoya, E., Munechika, D., Yang, H., Hoover, B., and Chau, D. H. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 893–911, 2023.

Wen, Y., Kirchenbauer, J., Geiping, J., and Goldstein, T. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *arXiv preprint arXiv:2305.20030*, 2023.

Wu, X., Liao, X., and Ou, B. Sepmark: Deep separable watermarking for unified source tracing and deepfake detection. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 1190–1201, 2023.

Xiong, C., Qin, C., Feng, G., and Zhang, X. Flexible and secure watermarking for latent diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 1668–1676, 2023.

Zhang, K. A., Xu, L., Cuesta-Infante, A., and Veeramachaneni, K. Robust invisible video watermarking with attention. *arXiv preprint arXiv:1909.01285*, 2019.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Zhu, J., Kaplan, R., Johnson, J., and Fei-Fei, L. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 657–672, 2018.

A. Compare of Watermarking Paradigms for Diffusion Models

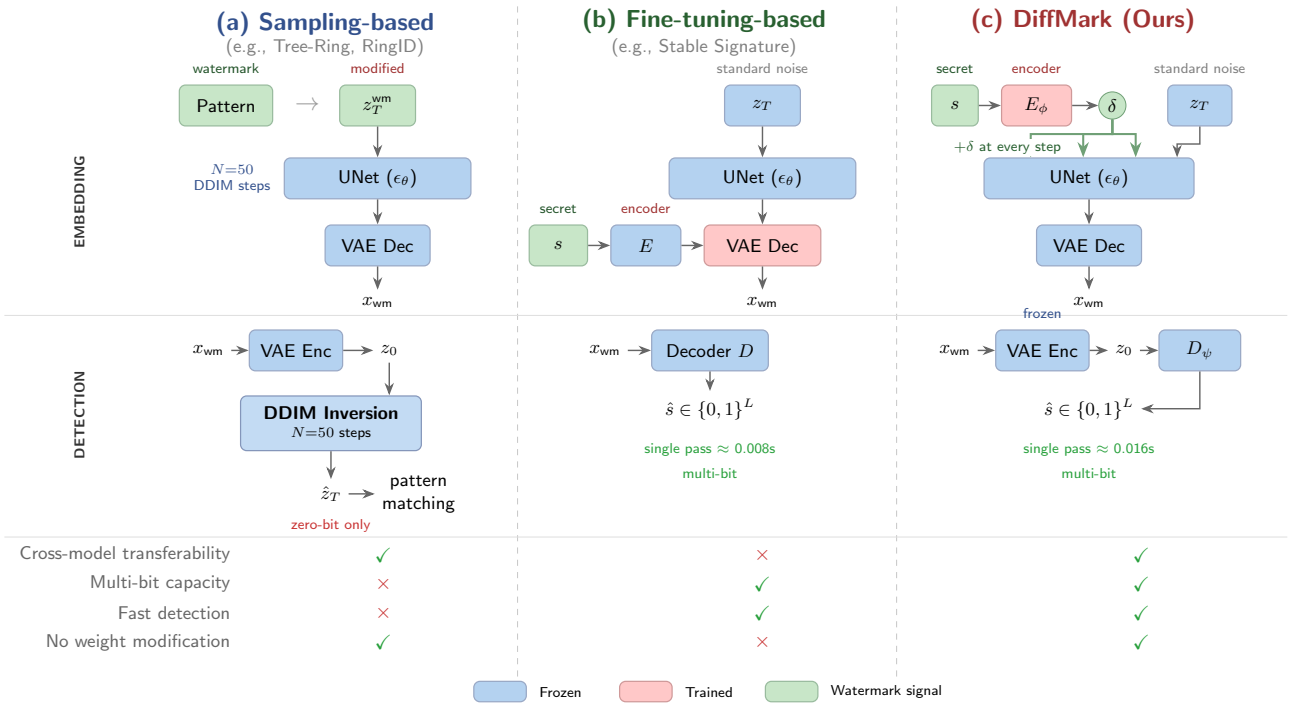


Figure 8. Comparison of watermarking paradigms for DMs. (a) Sampling-based methods, (b) Fine-tuning-based methods modify model weights for fast multi-bit detection, and (c) DiffMark (Ours) injects a learned perturbation δ at every step of a frozen UNet, enabling fast single-pass multi-bit detection without weight modification.

B. Detailed Related Work

B.1. Post-hoc Image Watermarking

Post-hoc watermarking methods embed imperceptible signals into *existing* images, regardless of how they were generated. Classical approaches modify coefficients in transform domains, such as Discrete Cosine Transform (DCT), Singular Value Decomposition (SVD) (Al-Haj, 2007; Navas et al., 2008). However, these methods are sensitive to compression algorithms and image editing techniques encountered in modern image-sharing platforms. Additionally, encryption schemes are also employed to prevent the removal of watermarks from adversarial (Neekhara et al., 2022; Wang et al., 2024).

Deep learning has enabled a more powerful paradigm: jointly training an encoder–decoder pair end-to-end with differentiable noise layers to learn robust embeddings. HiDDeN (Zhu et al., 2018) introduces this *encoder–noise layer–decoder* architecture, where differentiable noise layers inserted between the encoder and decoder during training force the model to learn robust encodings against Gaussian blur, cropping, and JPEG compression. MBRS (Jia et al., 2021) improves JPEG robustness by alternating between real and simulated compression across mini-batches. StegaStamp (Tancik et al., 2020) significantly raises the robustness bar by targeting *physical-world* distortions: it trains with aggressive augmentations simulating the print-and-photograph pipeline, including perspective warping via spatial transformer networks. RivaGAN (Zhang et al., 2019) introduces an adversarial training strategy where a dedicated attack network attempts to remove the watermark during training, pushing the encoder to find more resilient embedding strategies. This adversarial formulation was further developed by subsequent works that incorporated GAN-based discriminators to simultaneously improve watermarked image quality (Bui et al., 2023). SepMark (Wu et al., 2023) re-constructs this encoder–noise layer–decoder architecture into a novel deep separable watermarking framework that employs a single encoder alongside two distinct decoders: one robust and one semi-robust. Rather than embedding the watermark into the pixel-space, Fernandez (Fernandez et al., 2022) proposed watermarking in the latent space of self-supervised networks (e.g., DINO). This approach is resolution-agnostic and benefits from the transformation-invariant representations learned by self-supervised models, but the iterative per-image optimization makes embedding slower. More recently, the Watermark Anything Model (WAM) (Sander et al., 2025) reformulated watermarking as a segmentation task for localized multi-message extraction.

However, there are two main limitations of these post-hoc methods. First, since these methods operate on high-resolution pixel representations ($512 \times 512 \times 3$, they introduce substantial computational overhead and latency. Second, applying watermarks post-hoc in pixel space can introduce visual artifacts into the generated images.

B.2. Watermarking Diffusion Models

Methods that embed watermarks during the diffusion generation process fall into two families.

Sampling-based methods modify the sampling noise while keeping model weights frozen. Tree-Ring (Wen et al., 2023) pioneers this direction by embedding a concentric Fourier-space pattern into z_T. At detection time, the pattern is recovered by running a 50-step DDIM inversion, yielding a zero-bit (present/absent) decision. RingID (Ci et al., 2024) extends Tree-Ring to multi-key identification by assigning distinct ring patterns to different users, though detection still requires full inversion. Shallow Diffuse (Li et al., 2025) improves robustness by projecting the watermark to a low-dimensional subspace of z_T that is less affected by the denoising trajectory, demonstrating resilience against regeneration attacks. ROBIN (Huang et al., 2024) departs from modifying z_T entirely and instead implants a watermark at an intermediate diffusion state, using adversarial prompt optimization to hide the signal during subsequent denoising. Despite their plug-and-play nature, all these methods rely on DDIM inversion for detection, which requires $\sim$50 sequential UNet evaluations and is therefore prohibitive at platform scale.

Fine-tuning-based methods modify model components, most commonly the VAE decoder or the UNet, so that every generated image inherently carries an extractable watermark. Stable Signature (Fernandez et al., 2023) fine-tunes the VAE decoder jointly with a pre-trained extractor network, enabling multi-bit extraction in a single forward pass through the lightweight decoder without any inversion. AquaLoRA (Feng et al., 2024) targets the white-box setting where adversaries have full model access. The method merges watermark information directly into the UNet via a LoRA module, making the watermark inseparable from the model weights and resilient to module removal. Xiong (Xiong et al., 2023) propose embedding the secret within the latent decoder with flexible capacity control. While these methods achieve fast, multi-bit detection, they fundamentally couple the watermark to a specific model checkpoint: each model variant or fine-tuned derivative requires its own watermarking procedure.

Positioning of DiffMark. As summarized in Fig. 8, DiffMark bridges the two families. Like sampling-based methods, it operates on a frozen, unmodified diffusion model and requires no weight modification, making it truly plug-and-play across any LCM-compatible architecture. Like fine-tuning-based methods, it enables single-pass, multi-bit detection via a lightweight learned decoder, avoiding the costly inversion bottleneck. The key enabler is using LCMs as a differentiable training bridge: rather than embedding information in z_T (which forces inversion for recovery) or fine-tuning model weights (which couples the watermark to a specific checkpoint), DiffMark injects a learned perturbation δ at every denoising step of the frozen model, allowing the watermark signal to accumulate in z_0 where it can be directly extracted.

C. Details in Encoder-Decoder Pretraining

C.1. Encoder Architecture

Each bit position i is associated with two learnable embeddings in a table $\mathbf{W} \in \mathbb{R}^{2L \times d_e}$, one for each binary value. Given a secret s, the encoder retrieves the appropriate embedding for every bit and sums them into a single vector $\mathbf{x} = \sum_i \mathbf{W}[2i + s_i] \in \mathbb{R}^{d_e}$. This aggregated vector is then used to modulate a learned spatial basis $\mathbf{B} \in \mathbb{R}^{d_e \times h \times w}$ via an outer product:

$$\mathbf{X}_{d,i,j} = \sum_{d'} \mathbf{x}_{d'} \cdot \mathbf{B}_{d',i,j},$$

which lifts the secret representation to the full spatial resolution of the latent space. The resulting feature map is refined by three convolutional blocks (**Conv2d(3×3)** + SiLU + BN, channels: $d_e \to 32 \to 16 \to 8$), after which two parallel 3×3 convolution heads produce the parameters of a diagonal Gaussian: $\mu \in \mathbb{R}^{c \times h \times w}$ and $\log \sigma^2$. A learnable scalar α (initialized to 0.1) globally scales the output, providing a single knob to control perturbation strength independently of the learned feature magnitudes. The complete encoder contains 295,265 parameters (Tab. 3).

C.2. Decoder Architecture

The decoder D_ψ operates on the final denoised latent $z_0 \in \mathbb{R}^{4 \times h \times w}$, which already carries the accumulated watermark signal. Its design follows a standard classification backbone: a convolutional feature extractor followed by a fully connected

Table 3. Parameter breakdown of the encoder and decoder.

Module	Component	Parameters
Encoder E_ϕ	Bit embeddings ($2L \times d_e$)	8,192
	Spatial basis $\mathbf{B}$ ($d_e \times h \times w$)	262,144
	Refinement convolutions	$\sim$24K
	Variational heads (μ, $\log \sigma^2$)	$\sim$580
	Total	**295,265**
Decoder D_ψ	Input conv + 5 downsample blocks	$\sim$700K
	MLP head	$\sim$1.64M
	Total	**2,339,704**

classification head.

The feature extractor begins with a 3×3 input convolution ($4 \to 8$ channels, Sigmoid Linear Unit (SiLU)) and proceeds through five strided downsample blocks (**Conv2d(4×4, stride 2)** + SiLU + BN), progressively increasing channels ($8 \to 16 \to 32 \to 64 \to 128 \to 256$) while reducing spatial resolution from 64×64 to 2×2. BatchNorm is omitted from the final block to preserve representational flexibility near the classification boundary.

The resulting $256 \times 2 \times 2$ tensor is flattened to a 1,024-dimensional vector and processed by a three-layer MLP ($1024 \to 1024 \to 512 \to 2L$, with SiLU activations between layers). The output is reshaped to $\mathbb{R}^{L \times 2}$, providing per-bit logits from which the secret is recovered as $\hat{s}_i = \arg\max_c \text{logits}_{i,c}$. The decoder totals 2,339,704 parameters, with the MLP head accounting for the majority ($\sim$1.64M).

D. Details in Curriculum Training

This appendix provides the formal schedule definitions and gradient-level analysis from Sec. 3.3.

D.1. Delta Annealing Schedule

The magnitude target σ_{target} in $\mathcal{L}_{\text{mag}}$ (Eq. (9)) follows a cosine schedule that transitions from a relaxed initial value to a tighter final value:

$$\sigma_{\text{target}}(t) = \sigma_{\text{s}} + \frac{\sigma_{\text{e}} - \sigma_{\text{s}}}{2}\Big(1 - \cos\big(\pi \cdot \min(t/T_a,\, 1)\big)\Big), \tag{19}$$

where $\sigma_{\text{s}} > \sigma_{\text{e}} > 0$ are the initial and final targets and T_a is the annealing horizon. At $t = 0$, the target equals σ_{s}, assigning the encoder a large perturbation budget for establishing a decodable signal. As training progresses, the target smoothly decreases to σ_{e}, enforcing imperceptibility. The cosine shape provides a gradual transition in the middle of training, avoiding abrupt changes in constraints that could destabilize optimization.

D.2. Gradient Analysis of Training Collapse

We provide a more detailed analysis of the optimization failure mode that motivates the curriculum. Consider the encoder gradient at initialization ($t = 0$) when all losses are active simultaneously. The two dominant terms contributing gradients to E_ϕ are the LCM reconstruction loss and the latent fidelity loss:

$$\nabla_\phi \mathcal{L}(0) \;\supset\; \underbrace{\nabla_\phi \mathcal{L}_{\text{lcm}}}_{\text{requires } \|\delta\|>0} + w_{\text{lafid}} \cdot \underbrace{\nabla_\phi \mathcal{L}_{\text{lafid}}}_{\text{drives } \delta \to 0}. \tag{20}$$

The reconstruction gradient $\nabla_\phi \mathcal{L}_{\text{lcm}}$ flows through the chain $\mathcal{L}_{\text{lcm}} \to D_\psi \to z_0^{\text{lcm}} \to \text{LCM} \to \delta \to E_\phi$. At initialization, the randomly initialized decoder D_ψ produces near-uniform predictions regardless of z_0, yielding $\nabla_{z_0} \mathcal{L}_{\text{lcm}} \approx 0$. Consequently, the useful gradient signal reaching E_ϕ is negligible.

In contrast, the latent fidelity gradient $\nabla_\phi \mathcal{L}_{\text{lafid}} = \nabla_\phi \text{MSE}(z_0^{\text{lcm}}, z_0^{\text{lcm,clean}})$ is well-defined from the first step: any nonzero δ produces a nonzero $z_0^{\text{lcm}} - z_0^{\text{lcm,clean}}$, providing a strong gradient that pushes $\delta \to 0$. The PRVL and frequency losses exhibit analogous behavior. The resulting gradient imbalance causes $\|\delta\|$ to collapse before the decoder can learn to exploit the watermark signal, leading to the trivial solution $\delta = 0$.

The curriculum prevents this by activating $\mathcal{G}_{\text{imp}}$ only after the decoder has been sufficiently trained on $\mathcal{G}_{\text{rec}}$. At this point, D_ψ produces informative gradients $\nabla_{z_0}\mathcal{L}_{\text{lcm}} \neq 0$ that counterbalance the imperceptibility losses, enabling stable co-optimization of accuracy and quality.

E. Details in Imperceptibility Loss Functions

Latent fidelity loss $\mathcal{L}_{\text{lafid}}$.

$$\mathcal{L}_{\text{lafid}} = \text{MSE}\big(z_0^{\text{lcm}},\ z_0^{\text{lcm,clean}}\big), \tag{21}$$

where $z_0^{\text{lcm,clean}}$ is the LCM output from the same z_T with zero delta (detached from the encoder graph). This penalizes the global latent-space distortion introduced by δ before VAE decoding amplifies it into pixel-space artefacts. Weight w_{lafid}=0.1.

Peak regional variational loss $\mathcal{L}_{\text{prvl}}$. While $\mathcal{L}_{\text{lafid}}$ controls global distortion, the watermark energy may still concentrate in localized pixel patches. Following (Feng et al., 2024), we penalize the worst-case 32×32 regional mean absolute difference:

$$\mathcal{L}_{\text{prvl}} = \max_p\ \text{conv2d}\Big(\text{mean}_c\big[|x_{\text{wm}} - x_{\text{clean}}|\big],\ K_{32\times 32}\Big)_p, \tag{22}$$

where $K_{32\times 32} = \frac{1}{32^2}\cdot \mathbf{1}_{32\times 32}$ is a uniform averaging kernel. Minimising $\mathcal{L}_{\text{prvl}}$ forces the encoder to distribute watermark energy uniformly across the image plane. Gradients flow through VAE decoding and the LCM path back to E_ϕ. Weight w_{prvl}=1.5.

Frequency constraint $\mathcal{L}_{\text{freq}}$.

$$\mathcal{L}_{\text{freq}} = \frac{\text{mean}(P_{\text{low}})}{\text{mean}(P) + 10^{-8}}, \qquad P = |\mathcal{F}(\delta)|^2, \tag{23}$$

where $\mathcal{F}$ is the centered 2D FFT and P_{low} is the power inside a disk of radius 10 centered at DC. Penalizing the low-frequency energy ratio pushes δ toward high-frequency components (edges and textures), where the human visual system is least sensitive (Li et al., 2025). Weight w_{freq}=0.5; active from step 0.

F. Additional Implementation Details

Curriculum Schedule. Following the strategy in Sec. 3.3, the loss groups are activated at τ_{rec}=0 and τ_{imp}=500. The delta magnitude target σ_{target} is annealed from σ_s=0.10 to σ_e=0.05 over 5,000 steps via Eq. (19).

Training Hyperparameters. Table 4 lists all hyperparameters for training.

Table 4. Training hyperparameters.

Parameter	Value
Training steps	10,000
Batch size	16
Optimizer	AdamW
Learning rate (encoder / decoder)	5×10^{-5} / 3×10^{-4}
LR schedule	Warmup (500 steps) + linear decay to 10^{-6}
Gradient clip (encoder / decoder)	5.0 / 1.0
Precision	bf16 mixed-precision
LCM steps (K) / DDIM steps (N)	4 / 50
Guidance scale (w)	7.5
Curriculum gates	
τ_{rec} / τ_{imp}	0 / 500
Delta annealing (cosine)	
$\sigma_s \to \sigma_e$	$0.10 \to 0.05$
T_a	5,000

Loss Weights. Table 5 provides the complete loss weight configuration.

Evaluation. For each evaluation dataset (MS-COCO 2017 (Lin et al., 2014), DiffusionDB (Wang et al., 2023), DALL-E3 (An et al., 2024)), 1,000 prompts are randomly selected. For each prompt, a watermarked image is generated at 512×512 resolution using DDIM with N=50 steps. Given a test image x, detection proceeds in two steps: (i) encode to latent space, $z_0 = \mathcal{E}(x) \cdot f_s$; (ii) extract the secret, $\hat{s} = \arg\max D_\psi(z_0)$. Bit accuracy is Bit ACC $= 1 - \frac{1}{L}\sum_{i=1}^{L} \mathbb{1}[\hat{s}_i \neq s_i]$.

Table 5. Loss weights for main training.

Loss	Symbol	Weight
LCM reconstruction	w_{lcm}	1.0
DDIM supervision	w_{ddim}	1.0
Magnitude constraint	w_{mag}	5.0 (MSE variant)
KL regularization	β	$0.001 \to 0.05$ (warmup, 1K steps)
Orthogonality	w_{orth}	0.1
PRVL	w_{prvl}	1.5
Latent fidelity	w_{lafid}	0.1
Frequency constraint	w_{freq}	0.5
Negative entropy	w_{neg}	0.01
Regeneration (VAE round-trip)	w_{regen}	1.0

G. Additional Experiments and Analysis

G.1. Details about Attacks

We evaluate the robustness of DiffMark under 13 types of attacks, which are categorized into: distortion, regeneration, and adversarial attacks.

Distortion Attacks. We evaluate robustness under 8 distortion attacks spanning geometric transformations (rotation, resized crop, random erasing), photometric perturbations (brightness, contrast, additive Gaussian noise), and signal-level corruptions (Gaussian blur, JPEG compression). Each attack is parameterized by a severity level that ranges from benign to aggressive, as summarized in Table 6.

Table 6. Summary of distortion attacks used for robustness evaluation.

Attack	ID	Mechanism	Strength Range
Rotation	`Rotation`	Rotate by angle	$0^\circ \to 45^\circ$
Resized Crop	`RCrop`	Crop + resize	scale $1.0 \to 0.5$
Random Erasing	`Erase`	Zero-fill random patch	$0\% \to 25\%$ area
Brightness	`Bright`	Enhance brightness	factor $1.0 \to 2.0$
Contrast	`Contrast`	Enhance contrast	factor $1.0 \to 2.0$
Gaussian Blur	`Blur`	Gaussian blur kernel	size $0 \to 20$
Additive Noise	`Noise`	Gaussian noise $\mathcal{N}(0, \sigma^2)$	σ: $0.0 \to 0.1$
JPEG Compression	`Compress`	Lossy re-encoding	quality $90 \to 10$

Regeneration Attacks. Regeneration attacks aim to overwrite the watermark by re-encoding a watermarked image into a latent representation and reconstructing it through an alternative generative model. We adopt three variants from the WAVES benchmark (An et al., 2024). `Regen-VAE` passes the image through a pretrained compression VAE. `Regen-Diff` encodes the image into the latent space of a surrogate DM (Stable Diffusion v1.4), adds noise for a specified number of timesteps, and re-denoises, with the number of noising steps as the strength parameter. `Rinse-2xDiff` repeats this diffusive regeneration twice, achieving stronger watermark removal at the cost of greater quality degradation. These attacks are summarized in Table 7.

Adversarial Attacks. Adversarial attacks craft imperceptible perturbations to disrupt the watermark detection pipeline. We evaluate two embedding attacks from WAVES (An et al., 2024), which maximize the ℓ_2 divergence between the latent representation of the adversarial image and the original within an ℓ_∞ ball of radius ϵ, solved via PGD. `AdvEmbG-KLVAE8` uses the same KL-VAE encoder as the victim model (grey-box setting), while `AdvEmbB-RN18` targets a pretrained ResNet-18 encoder (black-box setting). The perturbation budget $\epsilon \in \{2/255, 4/255, 6/255, 8/255\}$ controls attack strength.

Table 7. Summary of regeneration and adversarial attacks used for robustness evaluation.

Attack	ID	Strength Range
Regeneration attacks		
VAE	`Regen-VAE`	quality $1 \rightarrow 7$
Diffusion	`Regen-Diff`	steps $40 \rightarrow 200$
Rinsing ($2\times$)	`Rinse-2xDiff`	steps $20 \rightarrow 100$
Adversarial attacks		
KL-VAE (grey-box)	`Adv-KLVAE8`	ϵ: $2/255 \rightarrow 8/255$
ResNet-18 (black-box)	`Adv-RN18`	ϵ: $2/255 \rightarrow 8/255$

G.2. Additional Clean Watermark Detection Results

To verify that the findings from Sec. 4.2 generalize beyond DiffusionDB, we evaluate all methods on two additional datasets: DALL-E3 (An et al., 2024) and MS-COCO 2017 (Lin et al., 2014) captions. Tab. 8 reports detection accuracy, generation consistency, and generation quality.

Three observations stand out. First, DiffMark consistently achieves higher per-bit accuracy than AquaLoRA on both datasets (0.9417 vs. 0.9124 on DALL-E3; 0.9407 vs. 0.9251 on MS-COCO) despite embedding a strictly longer secret (64 bits vs. 48 bits). Second, DiffMark attains a perfect TPR of 1.0 at 0.1% FPR on both datasets, whereas AquaLoRA drops to 0.9440 on DALL-E3, indicating that its shorter secret does not fully compensate for the lower per-bit accuracy under stringent false-positive constraints. Third, DiffMark preserves competitive generation quality: on MS-COCO it achieves the lowest FID (67.86) and CLIP-FID (5.74), outperforming all sampling-based baselines, and on DALL-E3, it obtains a CLIP-FID of 7.61, second only to RingID (7.29) while providing 64-bit multi-bit capacity that RingID lacks. These results confirm that the dual-path training strategy and persistent delta injection generalize across prompt distributions without degrading either detection reliability or perceptual quality.

Table 8. Quantitative comparison on DALL-E3 and MS-COCO. **Best** results among plug-and-play methods are **bolded**; <u>overall best</u> are underlined.

				Detection Accuracy		Generation Consistency		Generation Quality	
Method	Type	Plug &Play	Multi-bit	Bit Acc (Clean)	TPR@0.1%FPR (Clean)	PSNR↑	LPIPS↓	FID↓	CLIP-FID↓
DALL-E3									
StegaStamp	Post Generation	✗	100 bits	<u>0.9988</u>	1.0	9.48	0.7715	113.54	23.92
Stable Signature	Fine-tuning	✗	48 bits	0.9915	1.0	13.51	0.5812	96.51	9.43
AquaLoRA		✗	48 bits	0.9124	0.9440	<u>18.28</u>	<u>0.3120</u>	<u>70.00</u>	<u>4.98</u>
Tree-Ring	Sampling	✓	0 bit	—	1.0	9.74	0.7643	98.59	9.97
RingID		✓	11 bits	—	—	9.39	0.7715	**92.72**	**7.29**
Shallow Diffuse		✓	0 bit	—	1.0	9.70	0.7725	97.88	9.53
DiffMark (Ours)		✓	**64 bits**	**0.9417**	**1.0**	**9.75**	**0.7496**	93.86	7.61
MS-COCO									
StegaStamp	Post Generation	✗	100 bits	<u>0.9986</u>	1.0	8.79	0.7809	159.57	32.34
Stable Signature	Fine-tuning	✗	48 bits	0.9981	1.0	12.96	0.5282	65.13	5.96
AquaLoRA		✗	48 bits	0.9251	0.9940	<u>17.49</u>	<u>0.2872</u>	<u>46.87</u>	<u>3.61</u>
Tree-Ring	Sampling	✓	0 bit	—	1.0	8.98	0.7349	70.22	6.63
RingID		✓	11 bits	—	—	8.66	0.7405	102.34	10.33
Shallow Diffuse		✓	0 bit	—	1.0	8.89	0.7444	68.77	6.50
DiffMark (Ours)		✓	**64 bits**	**0.9407**	**1.0**	**8.91**	**0.7343**	**67.86**	**5.74**

G.3. Identification Analysis

G.3.1. Experimental Design

We generate 1,000 watermarked images from DiffusionDB prompts using SD v1.5, each embedded with an independently sampled random 64-bit secret $s_i \sim \text{Uniform}(\{0,1\}^{64})$. All images are generated at 512×512 resolution using DDIM

with N=50 steps. To evaluate identification across a wide range of deployment scales, we construct user databases of size $N \in \{10, 10^2, 10^3, 10^4, 10^5, 10^6\}$ for attacked scenarios and extend to $N \in \{10^7, 10^8\}$ for the clean setting. We employ a two-tier scaling strategy:

- **Tier 1** ($N \leq 1{,}000$): subsample N real keys from the pool of 1,000 ground-truth keys.
- **Tier 2** ($N > 1{,}000$): include all 1,000 real keys and fill the remaining $N - 1{,}000$ entries with random distractor keys sampled uniformly from $\{0,1\}^{64}$.

For each query image x_i with ground-truth key $s^{(i)}$, we: (1) encode it to the latent space $z_0 = \mathcal{E}(x_i) \cdot f_s$; (2) decode the secret $\hat{s}_i = \arg\max D_\psi(z_0)$; (3) compute the Hamming distance $d_H(\hat{s}_i, s^{(j)})$ to every key $s^{(j)}$ in the database; and (4) rank the correct key $s^{(i)}$ among all database entries by ascending Hamming distance. The image is correctly identified if its ground-truth key achieves rank 1 (i.e., has the smallest Hamming distance). We report *Top-1 identification accuracy*, the fraction of images whose ground-truth key is ranked first, as the primary metric. For each database size, we repeat the experiment over 10 independent trials with different random database compositions and report the mean. This accounts for variance introduced by the random distractor keys in Tier 2.

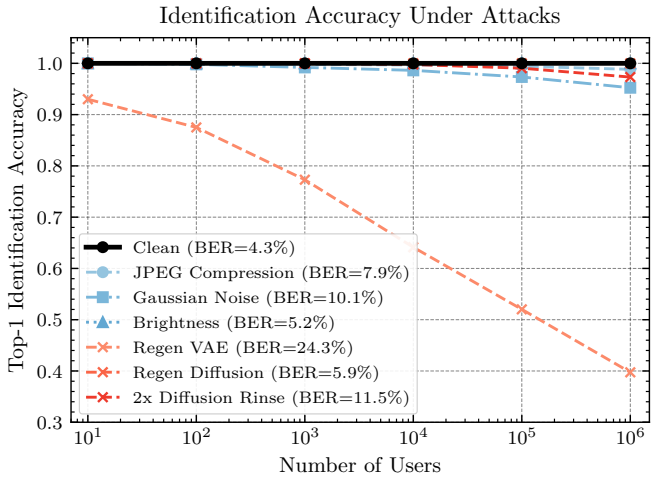


Figure 9. Identification accuracy under attacks. Top-1 accuracy vs. database size for representative distortion (solid) and regeneration (dashed) attacks. Most attacks preserve $> 95\%$ identification at 10^6 users.

G.3.2. Attack scenarios

Beyond the clean (no-attack) setting, we evaluate identification under 13 attack scenarios: 8 distortion attacks described in Sec. G.1. Attacked images are decoded using the same procedure, and the decoded bits are matched against the database. Fig. 9 extends this analysis to adversarial conditions, reporting Top-1 accuracy under representative distortion and regeneration attacks as a function of database size (N up to 10^6). Photometric distortions (brightness, contrast, JPEG compression, noise) and regeneration attacks (diffusion-based regen, 2× diffusion rinse) preserve near-perfect identification even at N=10^6, demonstrating that the watermark signal embedded by DiffMark is sufficiently robust for large-scale user attribution.

Only geometric attacks (rotation, resized crop, blur) degrade BER to $\sim$50%, collapsing identification, which is consistent with the detection-level vulnerabilities reported in Tab. 2.

G.4. Key Flexibility: Hamming Distance Analysis

Experimental Design Details. For each method, we generate two disjoint sets of 1,000 images using prompts sampled from DiffusionDB:

- **Fixed-key set.** All images are generated with the same predetermined secret s^* appeared during the training process.
- **Random-key set.** Each image i is assigned an independent secret $s_i \sim \text{Uniform}(\{0,1\}^L)$, sampled at generation time.

All images are generated at 512×512 resolution with SDv1.5. No post-processing or attack augmentation is applied. Each image is decoded by the corresponding method's detector to recover $\hat{s}$. We report: (1) **Per-image BER**: $\text{BER}_i = \frac{1}{L}\sum_{j=1}^{L} \mathbb{1}[\hat{s}_{i,j} \neq s_{i,j}]$; (2) **Mean and standard deviation of BER** across all 1,000 images per set.

Hamming Distance Analysis. To test whether decoding error depends on the proximity of a runtime secret to the training key s^*, we partition the random-key set into bins by Hamming distance $d_H(s_i, s^*)$ and compute mean BER per bin. We additionally compute the Pearson correlation r between $d_H(s_i; s^*)$ and BER_i over all 1,000 images. From Fig. 10, for both methods, Pearson correlations are statistically significant: DiffMark yields $r = -0.021$ ($p = 0.52$), while AquaLoRA yields $r = 0.028$ ($p = 0.37$). Binned analysis confirms this: for DiffMark, mean BER is 6.25%, 4.64%, and 4.55% across the low, mid, and high Hamming distance terciles, respectively; for AquaLoRA, the corresponding values are 31.25%, 28.17%, and 25.00%. The absence of a significant trend in either method indicates that both encoders generalize uniformly across the key space. This means that BER does not increase for keys distant from the training distribution center. This is a desirable

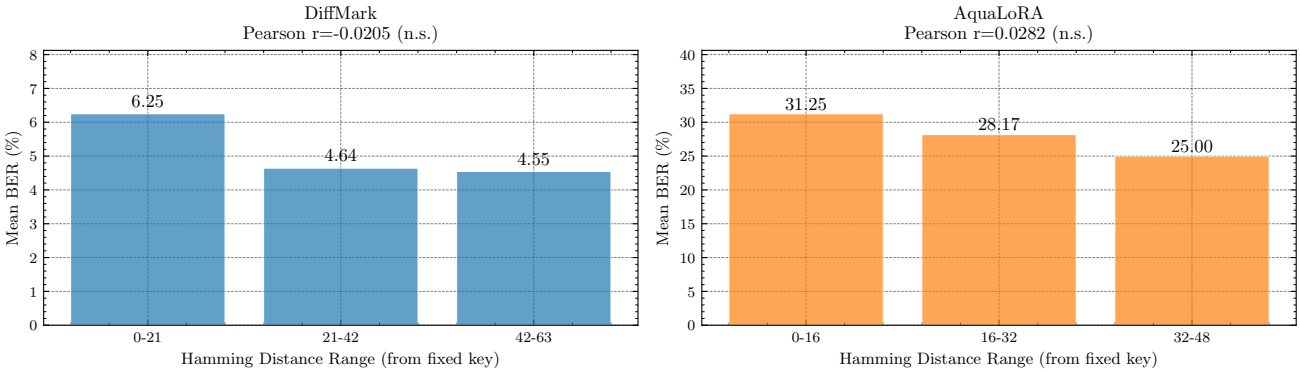


Figure 10. Per-image BER as a function of Hamming distance from the training key s^*.

property for practical multi-key deployment, as it ensures that any randomly sampled key achieves comparable decoding accuracy.

G.5. Cross-Model Transferability under Attacks

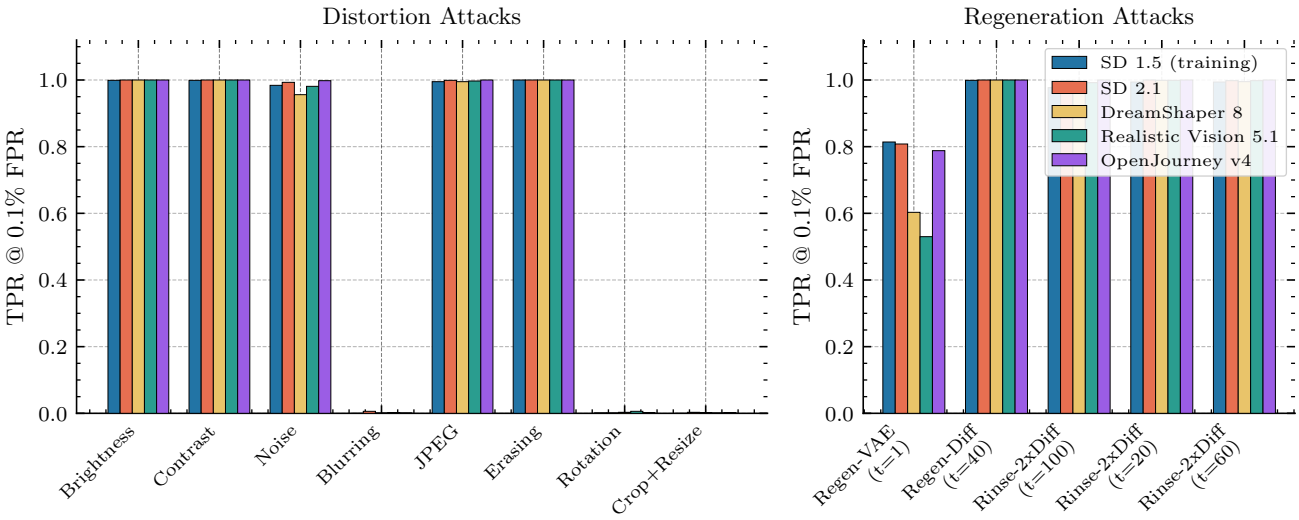


Figure 11. Cross-model transferability of DiffMark under attack. TPR@0.1%FPR is reported for five SD-family models across eight distortion attacks (left) and six regeneration attacks (right). DiffMark is trained exclusively on SD 1.5; no per-model fine-tuning is applied to the four unseen models.

Fig. 11 reveals two key results about DiffMark's cross-model robustness. First, the attack sensitivity profile of unseen models closely mirrors that of the SD 1.5 training model: all four target architectures achieve near-perfect TPR on photometric distortions (brightness, contrast, noise, JPEG, erasing) while scoring zero on geometric attacks (rotation, crop-resize), precisely replicating the pattern reported in Tab. 2. This consistency confirms that the robustness characteristics are inherited from the shared latent-space structure rather than being artifacts of the training model overfitting. Second, under regeneration attacks, there is a notable TPR drop under Regen-VAE for DreamShaper 8 and Realistic Vision 5.1. Future work can explore the reason behind this issue.

G.6. Cross-Model Transferability: Qualitative Results

Fig. 12 presents qualitative examples of DiffMark applied across five SD-family models, using identical prompts and watermark secrets. These results complement the quantitative findings in Sec. 4.4 by demonstrating that cross-model transferability incurs no perceptual cost.

G.7. Additional Robustness Results

Tab. 9 extends the robustness evaluation of Sec. 4.7 from DiffusionDB to DALL-E3 and MS-COCO, reporting TPR@0.1%FPR under the same 13 attack types. From this table, we can observe that:

- Robustness-oriented methods lead on average TPR. AquaLoRA and RingID, which explicitly target robustness during training, attain the highest average TPR on both additional datasets. On DALL-E3, AquaLoRA reaches 0.80 and RingID 0.75; on MS-COCO, AquaLoRA achieves 0.91 and StegaStamp 0.87.
- DiffMark is competitive without explicit robustness training. Although robustness is not the primary design focus of DiffMark, it still achieves an average TPR of 0.72 on DALL-E3 and 0.75 on MS-COCO. Second, persistent delta injection reinforces the watermark signal at every denoising step, making it difficult for regeneration attacks and black-box adversarial attacks to erase/overwrite the mark without substantially altering image content.
- However, the same geometric and frequency-domain vulnerabilities observed on DiffusionDB (rotation, blur, resized crop) persist on the additional datasets, as these attacks corrupt the latent encoding $z_0 = \mathcal{E}(x) \cdot f_s$ before the decoder can operate.

Table 9. Robustness comparison on DALL-E3 and MS-COCO under 13 attack types (TPR@0.1%FPR). **Best** results are **bolded**.

Attack	Type	StegaStamp	Stable Signature	AquaLoRA	Tree-Ring	RingID	Shallow Diffuse	**DiffMark (Ours)**
DALL-E3								
Bright	Distortion Attacks	**1.00**	**1.00**	0.89	0.64	**1.00**	**1.00**	**1.00**
Compress		**1.00**	0.98	0.93	0.78	**1.00**	**1.00**	**1.00**
Contrast		**1.00**	**1.00**	0.87	0.73	**1.00**	**1.00**	**1.00**
Erase		**1.00**	**1.00**	0.88	0.53	**1.00**	**1.00**	**1.00**
RCrop		0.37	**1.00**	0.82	0.03	0.00	0.00	0.00
Rotation		0.00	0.78	0.00	0.14	**1.00**	0.00	0.02
Blur		0.17	0.00	0.85	**1.00**	**1.00**	0.01	0.01
Noise		**1.00**	0.99	0.93	0.73	0.98	**1.00**	0.99
Regen-VAE	Regeneration Attacks	**1.00**	0.00	0.80	0.49	**1.00**	0.90	0.90
Regen-Diff		0.99	0.00	0.88	0.84	0.98	**1.00**	**1.00**
Rinse-2Xdiff		0.72	0.00	0.66	0.78	0.82	**0.99**	**0.99**
Adv-KLVAE8	Adversarial Attacks	**1.00**	**1.00**	0.90	0.85	0.00	0.60	0.50
Adv-RN18		**1.00**	**1.00**	0.94	**1.00**	0.00	**1.00**	**1.00**
Average		0.79	0.67	**0.80**	0.66	0.75	0.73	0.72
MS-COCO								
Bright	Distortion Attacks	**1.00**	**1.00**	0.99	0.76	**1.00**	**1.00**	**1.00**
Compress		**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**	**1.00**
Contrast		**1.00**	**1.00**	0.99	0.86	**1.00**	**1.00**	**1.00**
Erase		**1.00**	**1.00**	**1.00**	0.78	**1.00**	**1.00**	**1.00**
RCrop		0.42	**1.00**	0.99	0.09	0.05	0.02	0.01
Rotation		0.01	0.98	0.00	0.52	**1.00**	**0.03**	0.02
Blur		0.43	0.00	**0.99**	0.44	**0.99**	**0.03**	0.01
Noise		**1.00**	**1.00**	**1.00**	0.82	**1.00**	**1.00**	0.99
Regen-VAE	Regeneration Attacks	**1.00**	0.02	0.98	0.68	**1.00**	0.97	0.97
Regen-Diff		0.98	0.02	**1.00**	0.91	**1.00**	**1.00**	**1.00**
Rinse-2Xdiff		0.89	0.01	0.94	0.88	**1.00**	**0.99**	**1.00**
Adv-KLVAE8	Adversarial Attacks	**1.00**	**1.00**	0.99	0.50	0.55	0.72	0.74
Adv-RN18		**1.00**	**1.00**	**1.00**	0.91	0.67	**1.00**	**1.00**
Average		0.83	0.69	**0.91**	0.70	0.87	0.75	0.75

G.8. More Qualitative Results

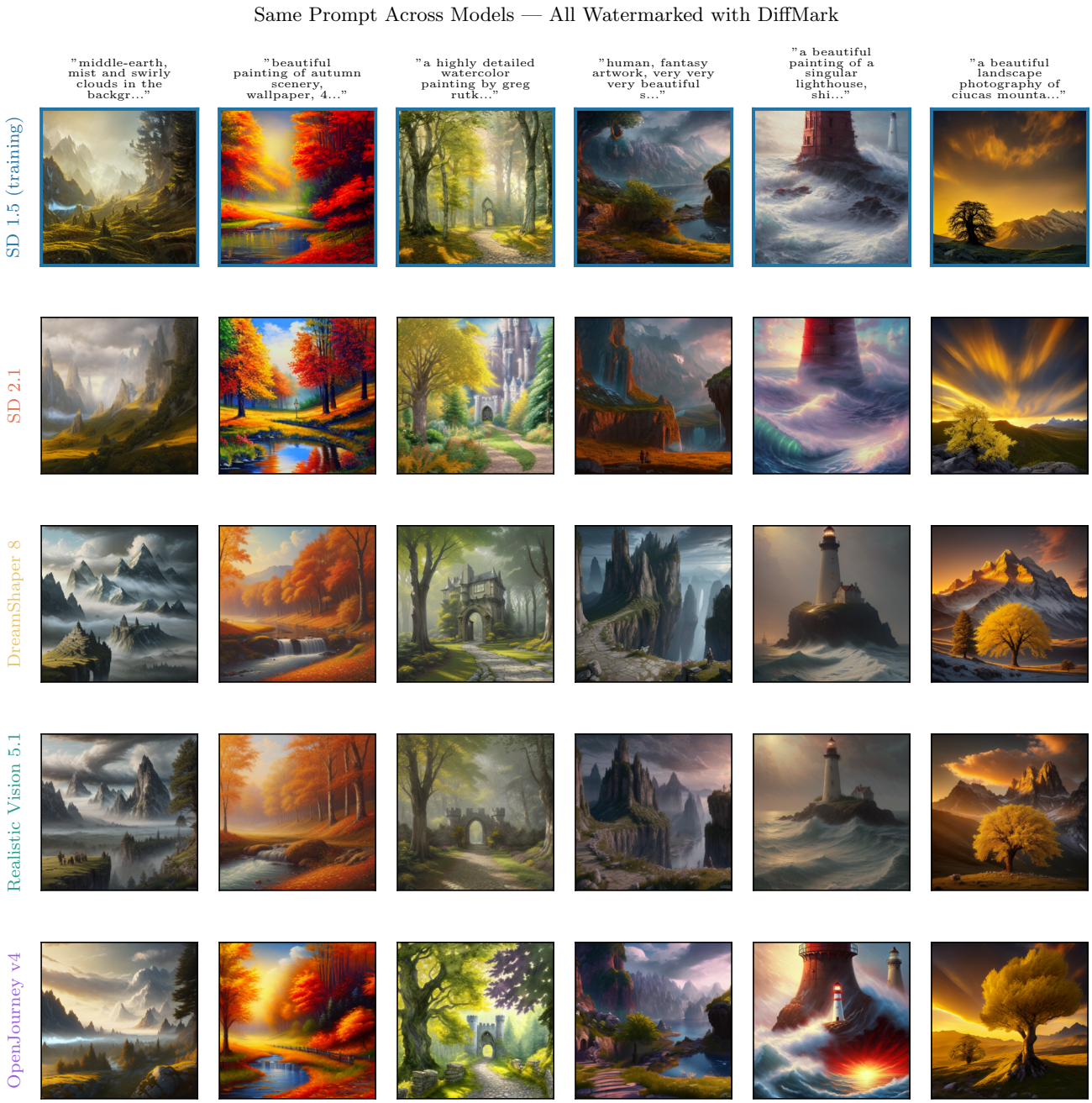
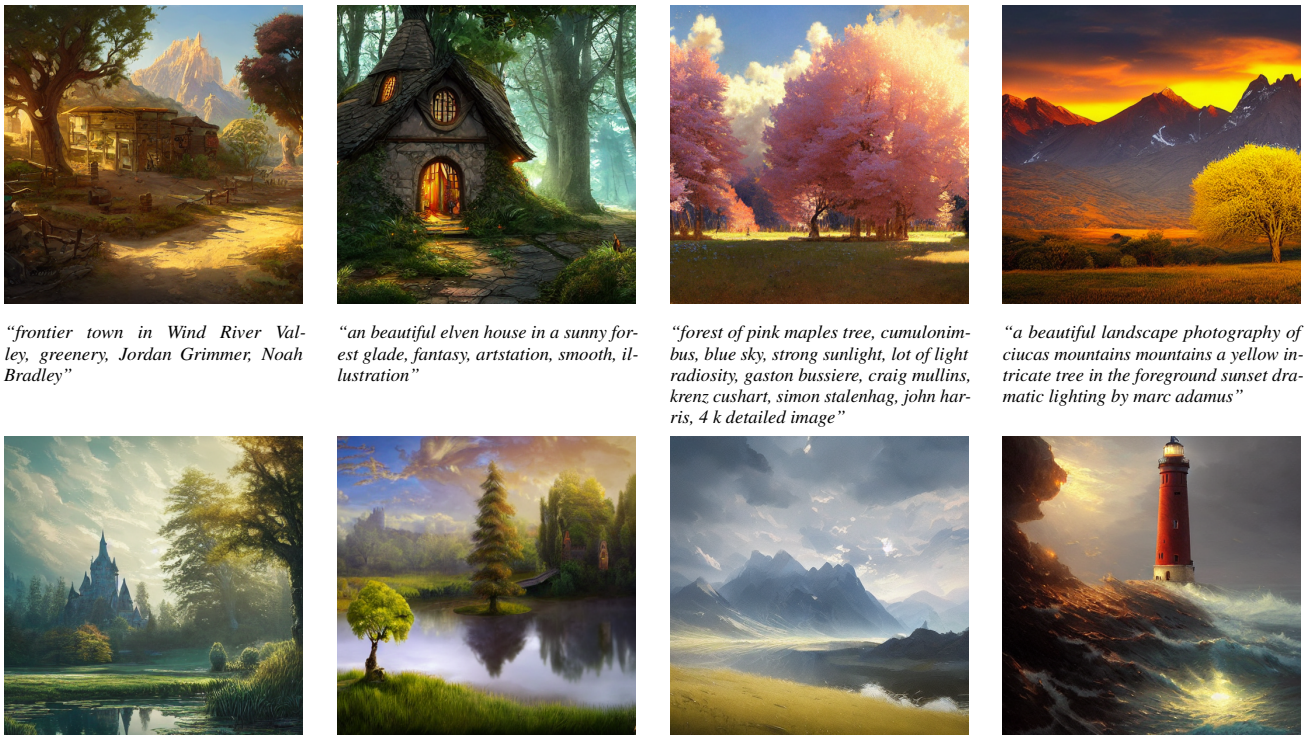


Figure 12. **Qualitative cross-model transferability of DiffMark.** Each column shares the same prompt; each row corresponds to a different SD-family model. DiffMark is trained exclusively on SD 1.5 and applied to four unseen models (SD 2.1, DreamShaper 8, Realistic Vision 5.1, OpenJourney v4) without any per-model fine-tuning.



"frontier town in Wind River Valley, greenery, Jordan Grimmer, Noah Bradley"

"an beautiful elven house in a sunny forest glade, fantasy, artstation, smooth, illustration"

"forest of pink maples tree, cumulonimbus, blue sky, strong sunlight, lot of light radiosity, gaston bussiere, craig mullins, krenz cushart, simon stalenhag, john harris, 4 k detailed image"

"a beautiful landscape photography of ciucas mountains mountains a yellow intricate tree in the foreground sunset dramatic lighting by marc adamus"

"a tree near a pond, a castle and mist and swirly clouds in the background, fantastic landscape, hyperrealism, no blur, 4k resolution, ultra detailed, style of Anton Fadeev, Ivan Shishkin, John Berkey, James Jean"

"a tree near a pond, a castle and mist and swirly clouds in the background, fantastic landscape, hyperrealism, no blur, 4k resolution, ultra detailed, style of Anton Fadeev, Ivan Shishkin, John Berkey"

"painting of a landscape, concept art, blurry, broad strokes, canvas, first light, majestic mountains, lake, lush grass, dramatic clouds, soft light, by greg rutkowski and jakub rozalski, lip comarella and eytan zana"

"a beautiful painting of a singular lighthouse, shining its light across a tumultuous sea of blood by greg rutkowski and thomas kinkade, trending on artstation"

Figure 13. Qualitative examples of watermarked images generated by DiffMark with their corresponding input prompts. The embedded watermark is visually imperceptible.

G.9. Watermark Signal Visualization

We visualize the learned watermark signal to provide qualitative insight into how DiffMark embeds information.

Signal analysis. 14 shows watermarked and clean images side-by-side with their amplified difference maps and latent delta heatmaps. The difference maps (pixel-space $|\mathbf{x}_{\text{wm}} - \mathbf{x}_{\text{clean}}|$, amplified $10\times$) reveal that the watermark concentrates along edges and textured regions, consistent with the high-frequency constraint $\mathcal{L}_{\text{freq}}$. The delta heatmaps show spatially uniform energy distribution, validating the PRVL regularizer (Sec. 3.2.3).

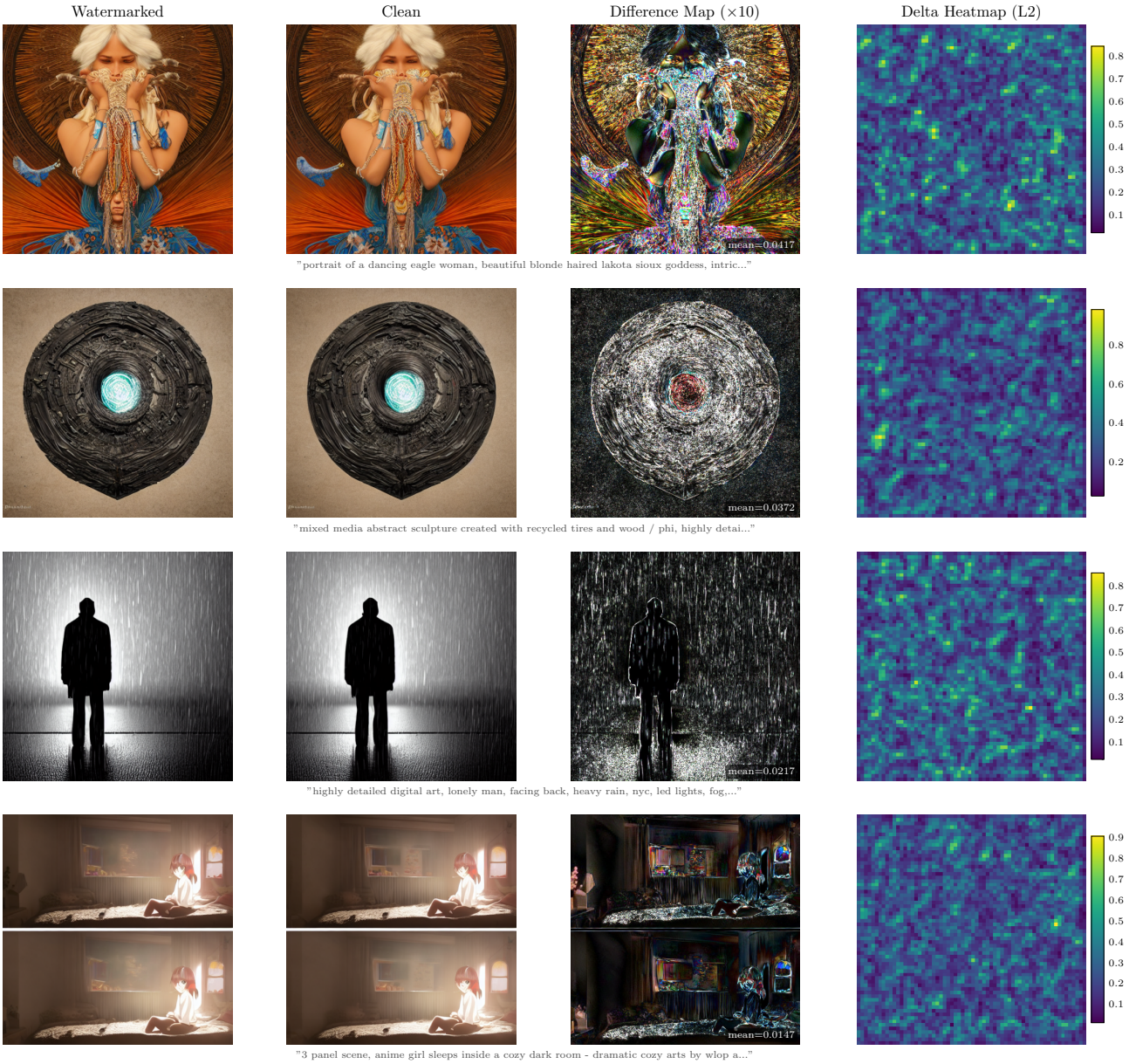


Figure 14. Watermark signal analysis. Columns from left to right: watermarked image, clean image (same $\mathbf{z}_T$ and prompt), difference map ($10\times$ amplified), and delta heatmap (L2 norm across latent channels). The watermark perturbation is imperceptible in pixel space while maintaining spatially uniform energy in latent space.

Training progression. 15 illustrates how the δ signal evolves during curriculum training. Before PRVL activation (step 439), the δ exhibits uneven spatial distribution. After PRVL (step 939), energy becomes uniformly distributed across the 64×64 latent grid. In the difference maps, the frequency constraint progressively steers perturbations away from smooth regions toward edges and textures, improving imperceptibility.

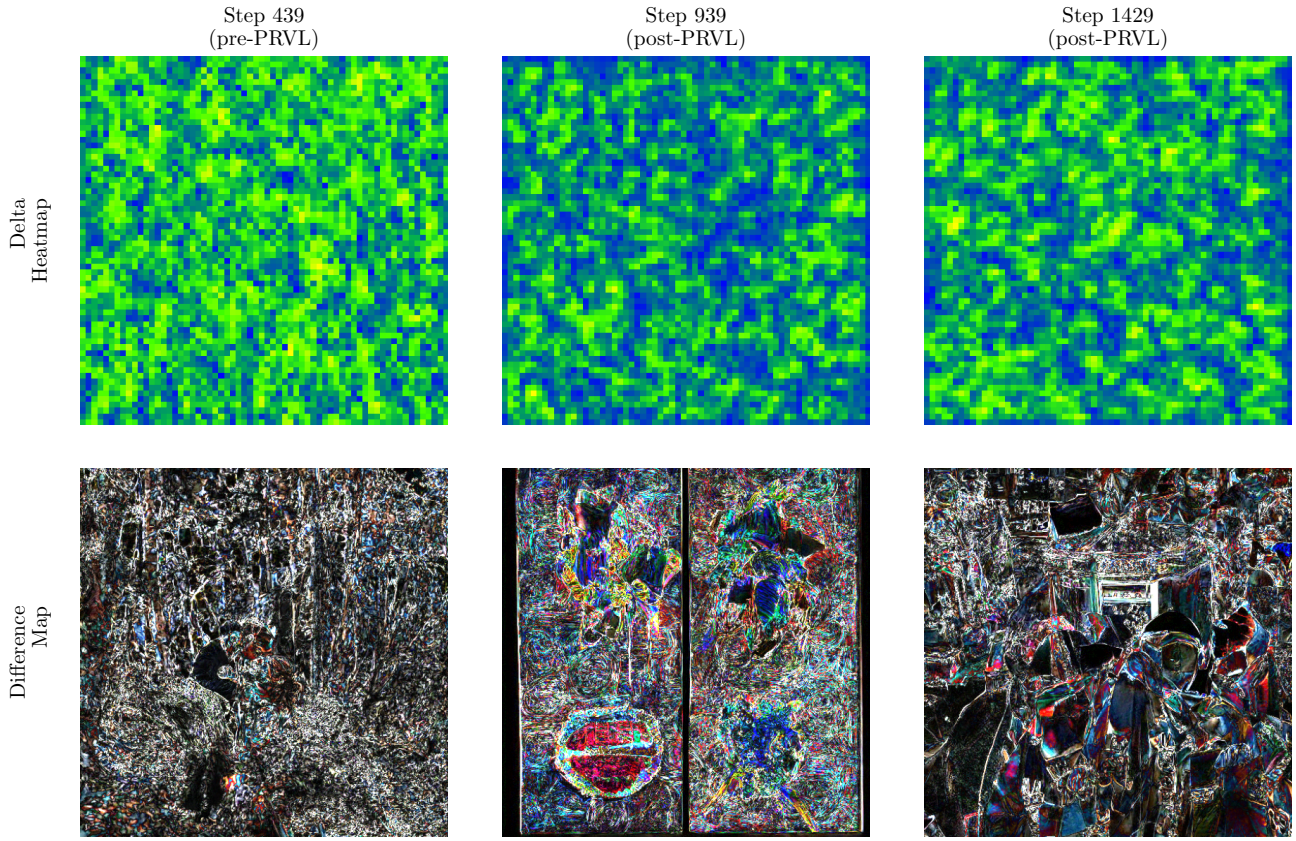


Figure 15. Training progression of the watermark signal. Top row: δ heatmaps; bottom row: difference maps. Left to right: early training (pre-PRVL), mid training (post-PRVL), and late training (near convergence). The PRVL regularizer enforces spatial uniformity in δ, while $\mathcal{L}_{\text{freq}}$ pushes pixel-space differences toward high-frequency regions.

Per-channel δ structure. 16 decomposes the learned $\boldsymbol{\delta} \in \mathbb{R}^{4\times64\times64}$ into its four latent channels. Each channel carries a distinct spatial pattern with both positive and negative perturbations, confirming that the encoder distributes the watermark signal across all latent dimensions rather than concentrating it in a single channel.

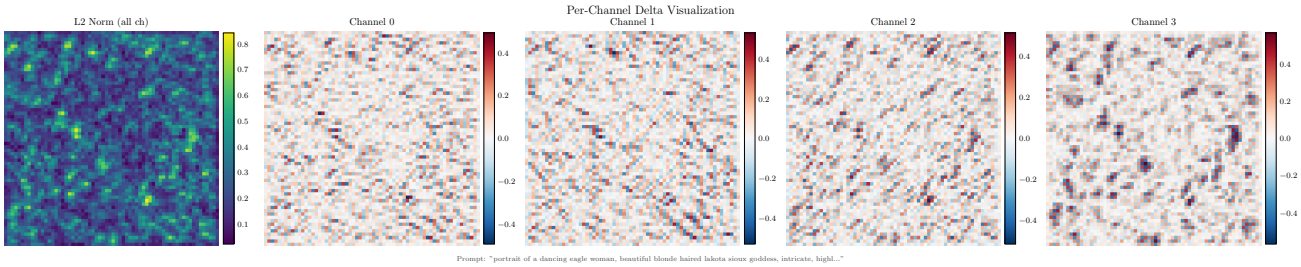


Figure 16. Per-channel δ decomposition. Left: overall L2 norm across channels. Right four panels: individual latent channels visualized with a diverging colormap. The watermark signal is distributed across all four channels with distinct spatial patterns.

Frequency spectrum analysis. 17 presents the 2D FFT power spectrum of the learned δ, validating the effect of the frequency constraint $\mathcal{L}_{\text{freq}}$. The radial power profile shows suppressed energy within the low-frequency radius ($r < 10$), confirming that the encoder has learned to avoid low-frequency perturbations that would be perceptually salient. The remaining energy is distributed across mid-to-high frequencies, which correspond to edges and fine textures in pixel space.

Difference map overlay. 18 overlays the pixel-space difference heatmap onto the watermarked images, revealing that the watermark perturbation concentrates at edges, contours, and high-texture regions. This spatial distribution aligns with human visual masking, where modifications in high-frequency regions are less perceptible.

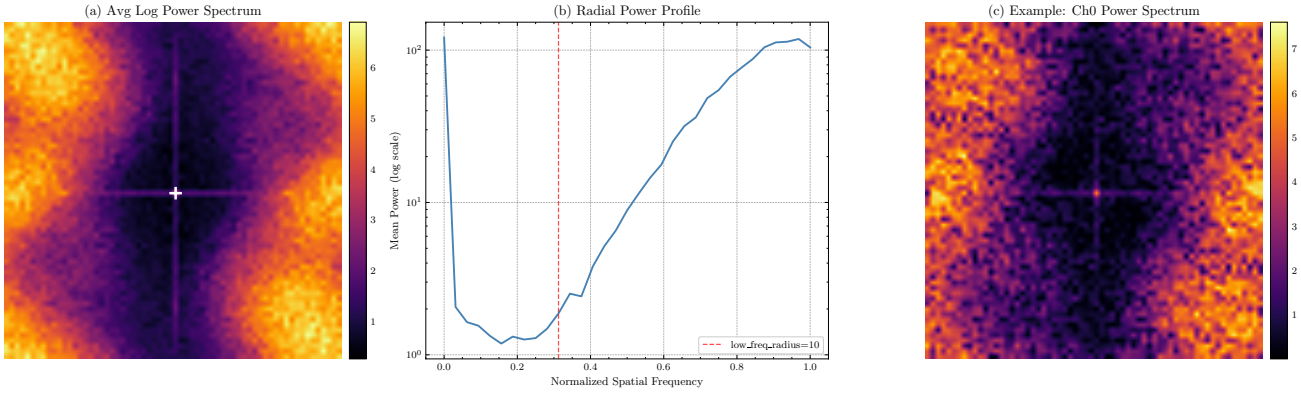


Figure 17. Frequency domain analysis of the δ signal. (a) Average log power spectrum across all samples and channels, with DC at center. (b) Radial power profile showing suppressed low-frequency energy below the configured radius (red dashed line). (c) Single-example power spectrum for channel 0.

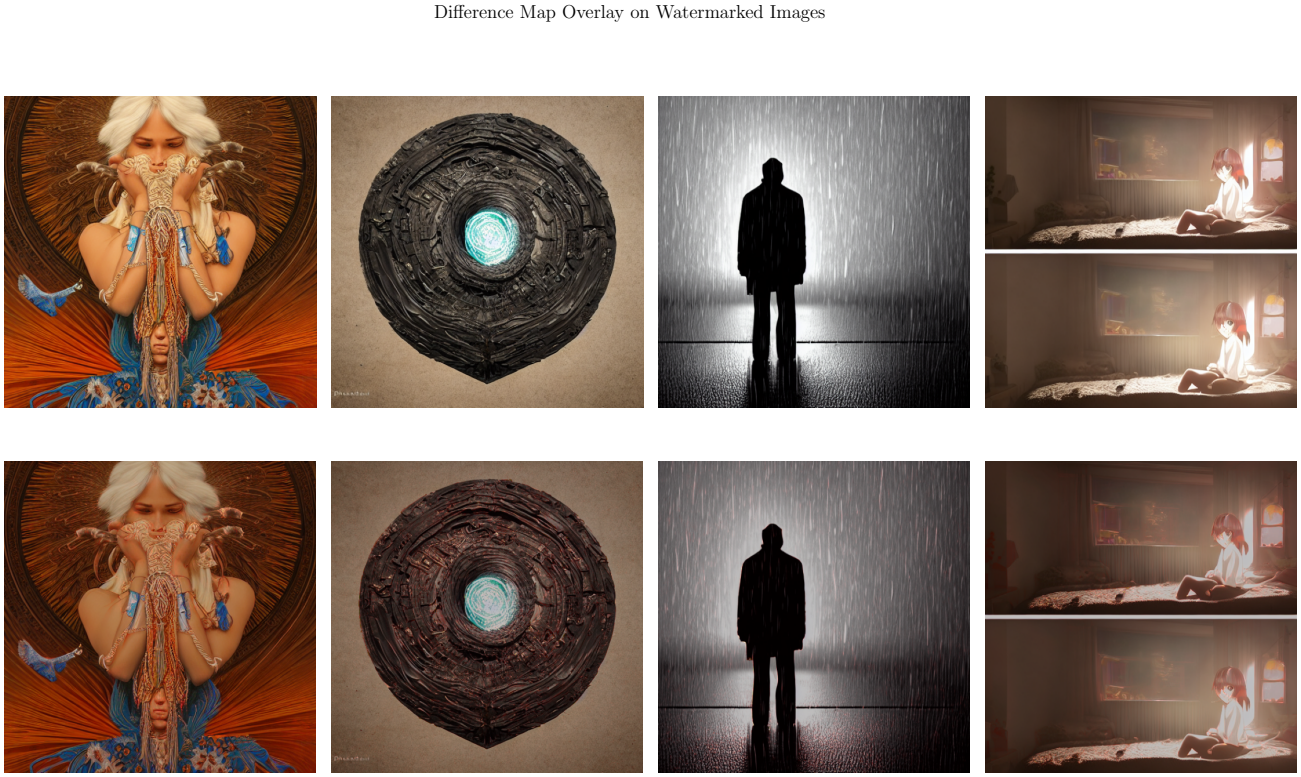


Figure 18. Difference map overlay on watermarked images. Top row: watermarked images. Bottom row: same images with the mean absolute difference rendered as a semi-transparent heatmap. Bright regions indicate stronger watermark perturbation, which concentrates at edges and textures.

H. Pseudo-codes

H.1. Encoder-Decoder Pretraining

Algorithm 1 Encoder-Decoder Pretraining

Require: Encoder E_ϕ, Decoder D_ψ, secret length L, pretraining steps N_{pre}, noise schedule $\sigma_{\text{start}} \to \sigma_{\text{end}}$
Ensure: Pretrained E_ϕ, D_ψ

for $t = 1, \ldots, N_{\text{pre}}$ **do**
  $s \sim \text{Bernoulli}(0.5)^L$
  $\delta \leftarrow E_\phi(s)$
  // Clean reconstruction
  $\mathbf{o}_{\text{clean}} \leftarrow D_\psi(\delta)$
  $\mathcal{L}_{\text{clean}} \leftarrow \mathcal{L}_{\text{CE}}(\mathbf{o}_{\text{clean}},\ s)$ {Eq. (11)}
  // Noisy reconstruction (curriculum)
  $\sigma_n \leftarrow \sigma_{\text{start}} + (\sigma_{\text{end}} - \sigma_{\text{start}}) \cdot t/N_{\text{pre}}$
  $\epsilon \sim \mathcal{N}(0, \sigma_n^2 \mathbf{I})$
  $\mathbf{o}_{\text{noisy}} \leftarrow D_\psi(\delta + \epsilon)$
  $\mathcal{L}_{\text{noisy}} \leftarrow \mathcal{L}_{\text{CE}}(\mathbf{o}_{\text{noisy}},\ s)$
  // Regularization
  $\mathcal{L}_{\text{orth}} \leftarrow \frac{1}{B(B-1)} \sum_{i \neq j} \frac{\langle \delta_i, \delta_j \rangle_F}{\|\delta_i\|_F \|\delta_j\|_F}$ {Eq. (12)}
  $\mathcal{L} \leftarrow w_r \cdot \mathcal{L}_{\text{clean}} + w_n \cdot \mathcal{L}_{\text{noisy}} + w_{\text{orth}} \cdot \mathcal{L}_{\text{orth}}$
  Update ϕ, ψ via AdamW on $\nabla \mathcal{L}$
  if clean accuracy ≥ 0.99 for 10 consecutive steps **then**
    break
  end if
end for

H.2. Dual-path Training

Algorithm 2. DiffMark Training with Dual-Path LCM Bridge

Require: Pretrained encoder E_ϕ, decoder D_ψ; frozen UNet ϵ_θ, LCM, VAE $(\mathcal{E}, \mathcal{D})$; curriculum gates $\tau_{\text{rec}}, \tau_{\text{imp}}, \tau_{\text{rob}}$; training steps T
Ensure: Trained E_ϕ, D_ψ

for $t = 1, \ldots, T$ **do**
  $s \sim \text{Bernoulli}(0.5)^L$; $z_T \sim \mathcal{N}(0, \mathbf{I})$; $c \leftarrow \text{CLIP}(\text{prompt})$
  $\delta \leftarrow E_\phi(s)$

  ***// LCM path** (differentiable, K=4 steps)*
  $z \leftarrow z_T$
  for $k = 1, \ldots, K$ **do**
    $\tilde{z} \leftarrow z + \delta$ {Delta injection (Eq. (5))}
    $z \leftarrow \text{LCM}_\theta(\tilde{z},\ t_k,\ c)$
  end for
  $z_0^{\text{lcm}} \leftarrow z$
  $\mathcal{L}_{\text{lcm}} \leftarrow \mathcal{L}_{\text{CE}}(D_\psi(z_0^{\text{lcm}}),\ s)$ {Eq. (14): ∇ to both E_ϕ and D_ψ}

  ***// DDIM path** (non-differentiable, N=50 steps)*
  $\bar{\delta} \leftarrow \text{sg}(\delta)$ {Stop gradient}
  $z \leftarrow z_T$
  for $k = 1, \ldots, N$ **do**
    $\tilde{z} \leftarrow z + \bar{\delta}/N$ {Scaled injection (Eq. (15))}
    $z \leftarrow \text{DDIM}_\theta(\tilde{z},\ t_k,\ c)$
  end for
  $z_0^{\text{ddim}} \leftarrow z$
  $\mathcal{L}_{\text{ddim}} \leftarrow \mathcal{L}_{\text{CE}}(D_\psi(z_0^{\text{ddim}}),\ s)$ {∇ to D_ψ only}

23:
24: ***// Curriculum-gated losses***
25: $\mathcal{L} \leftarrow \mathcal{L}_{\text{lcm}} + \mathcal{L}_{\text{ddim}} + w_{\text{mag}} \cdot \mathcal{L}_{\text{mag}}$
26: **if** $t \geq \tau_{\text{imp}}$ **then**
27: $z_0^{\text{clean}} \leftarrow \text{sg}\big(\text{LCM}(z_T, \mathbf{0}, c)\big)$
28: $\mathcal{L} \mathrel{+}= w_{\text{lafid}} \cdot \text{MSE}(z_0^{\text{lcm}}, z_0^{\text{clean}}) + w_{\text{prvl}} \cdot \mathcal{L}_{\text{prvl}} + w_{\text{freq}} \cdot \mathcal{L}_{\text{freq}}$
29: *// VAE round-trip robustness*
30: $z_0^{\text{regen}} \leftarrow \mathcal{E}\big(\text{clamp}(\mathcal{D}(z_0^{\text{ddim}}/f_s),\, -1,\, 1)\big) \cdot f_s$
31: $\mathcal{L} \mathrel{+}= w_{\text{neg}} \cdot \mathcal{L}_{\text{neg}} + w_{\text{regen}} \cdot \mathcal{L}_{\text{CE}}(D_\psi(z_0^{\text{regen}}), s)$
32: **end if**
33:
34: ***// Update***
35: $\mathcal{L}$.backward()
36: Clip ∇_ϕ to norm 5.0; clip ∇_ψ to norm 1.0
37: Update ϕ, ψ via AdamW
38: $\sigma_{\text{target}} \leftarrow \text{CosineAnneal}(t, \sigma_s, \sigma_e, T_a)$
39: **end for**

H.3. Watermark Embedding (Inference)

Algorithm 3 Watermark Embedding (Inference)

Require: Secret $s \in \{0,1\}^L$, text prompt p, trained encoder E_ϕ, frozen UNet ϵ_θ, VAE decoder $\mathcal{D}$, DDIM steps N, guidance scale w
Ensure: Watermarked image x_{wm}
1: $\delta \leftarrow E_\phi(s)$
2: $z_T \sim \mathcal{N}(0, \mathbf{I})$
3: $c \leftarrow \text{CLIP}(p)$
4: $z \leftarrow z_T$
5: **for** $k = 1, \ldots, N$ **do**
6: $\tilde{z} \leftarrow z + \delta$ {Persistent delta injection}
7: $\hat{\epsilon} \leftarrow (1+w)\,\epsilon_\theta(\tilde{z}, t_k, c) - w\,\epsilon_\theta(\tilde{z}, t_k, \varnothing)$ {CFG}
8: $z \leftarrow \sqrt{\bar{\alpha}_{t_{k+1}}} \left(\frac{\tilde{z} - \sqrt{1-\bar{\alpha}_{t_k}}\,\hat{\epsilon}}{\sqrt{\bar{\alpha}_{t_k}}} \right) + \sqrt{1-\bar{\alpha}_{t_{k+1}}}\,\hat{\epsilon}$ {DDIM step}
9: **end for**
10: $z_0 \leftarrow z$
11: $x_{\text{wm}} \leftarrow \mathcal{D}(z_0/f_s)$
12: **return** x_{wm}

H.4. Watermark Detection (Inference)

Algorithm 4 Watermark Detection (Inference)

Require: Test image x, trained decoder D_ψ, VAE encoder $\mathcal{E}$, scaling factor f_s, detection threshold τ
Ensure: Recovered secret $\hat{s}$, detection decision

$z_0 \leftarrow \mathcal{E}(x) \cdot f_s$ {Encode to latent space}
$\mathbf{o} \leftarrow D_\psi(z_0) \in \mathbb{R}^{L \times 2}$ {Single forward pass}
$\hat{s}_i \leftarrow \arg\max_{c \in \{0,1\}} \mathbf{o}_{i,c}$ for $i = 1, \ldots, L$ {Per-bit hard decision}
// Hypothesis test (Sec. G.2)
if registered secret s^* is provided **then**
  $m \leftarrow \sum_{i=1}^{L} \mathbb{1}[\hat{s}_i = s_i^*]$ {Matching bits}
  if $m > \tau$ **then**
    return $\hat{s}$, WATERMARKED
  else
    return $\hat{s}$, NOT WATERMARKED
  end if
end if
return $\hat{s}$