ERASE TO FORGE, VERIFY TO DEFEND: DIFFUSION-BASED FORGERY ATTACK AND SELF-VERIFIABLE DEFENSE MECHANISM

Anonymous authors
Paper under double-blind review

ABSTRACT

Watermarking has emerged as a critical mechanism for establishing provenance of AI-generated media, yet its security against forgery attacks remains largely underexplored. Forgery attacks, wherein an adversary embeds a counterfeit watermark into arbitrary content to falsely attribute it to a target model, pose a severe threat to the integrity of watermarking systems. In this work, we introduce **EraseForge**, a diffusion-based no-box forgery attack that repurposes watermark removal techniques to construct a proxy dataset of watermark-free images, from which a conditional diffusion model learns to estimate and reproduce the target watermark residual. By modeling the watermark as a structured residual rather than regenerating the full image, EraseForge achieves state-of-the-art forgery performance with as few as 1,000 training samples—a $5\times$ improvement in data efficiency over existing methods. We further demonstrate that EraseForge circumvents the multi-message defense strategy previously proposed in the literature, achieving over 91% bit accuracy at $N{=}20{,}000$ samples where prior methods fail at $N{=}50{,}000$. To address this threat, we propose a *self-verifiable message* defense mechanism with two complementary variants—SeedBind and ContentBind—that transform verification from a finite message pool lookup into a structural consistency check, reducing forged bit accuracy to $\sim 50\%$ (near-random) across all tested watermarking schemes and attack strategies without modifying the underlying watermarking architecture. Specifically, SeedBind enforces deterministic consistency between a random seed and its pseudo-random expansion to neutralize estimator-based attacks, while ContentBind derives the watermark message directly from the image's learned visual features, additionally neutralizing residual-based one-shot forgery attacks.

1 INTRODUCTION

The rapid proliferation of high-fidelity generative models (Gong et al., 2025; Deepmind, 2025; OpenAI) has necessitated robust mechanisms for digital provenance to mitigate the risks of misinformation and intellectual property theft. As synthetic media becomes increasingly indistinguishable from authentic content, watermarking has emerged as a primary technical solution for establishing provenance of generative media (Deepmind, 2023; Gowal et al., 2025; Amazon, 2024). This shift is driven not only by security concerns but also by an evolving global regulatory landscape such as the introduction of EU AI act (European Parliament and Council of the European Union, 2024), U.S. Executive Orders (The White House, 2023) and California AI Transparency Act (SB 942) (California State Legislature, 2024). These frameworks transition watermarking from a discretionary feature to a critical compliance requirement for model providers.

Current research in the security of watermarking has primarily focused on two adversarial vectors: Removal and Forgery. Although the former received considerable interest within research (Zhao et al., 2024; Liu et al., 2025), the latter remains largely unexplored. Forgery attacks, where an adversary creates a counterfeit watermark and applies it to new media, arguably pose a greater systematic threat to model providers (Dong et al., 2025). While removal attacks allow a user to use a model's output anonymously, forgery attacks allow an adversary to impersonate the model provider thus allowing the attacker to falsely attribute harmful media to a specific AI service. This weaponization of watermarks

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

erodes the "root of trust", potentially exposing service providers to severe reputation damage, legal liability, and regulatory fines.

Existing forgery attacks can be categorized along two axes: *adversarial access* and *underlying approach*. By adversarial access they divide into: *white-box attacks* (Kutter et al., 2000; Kinakh et al., 2024), which exploit knowledge of the embedding scheme; *black-box methods* (Wang et al., 2021; Müller et al., 2024), which query the watermarking or detection API with the objective of collecting watermark and non-watermark images; and *no-box methods* (Yang et al., 2024a; Souček et al., 2025b; Dong et al., 2025), which operate solely from a collection of watermarked images. Since service providers rarely disclose their watermarking algorithm, the no-box setting best reflects practical threat conditions; we therefore focus on this regime. Orthogonally, by underlying approach, forgery attacks—particularly no-box ones—split into *residual-based* and *estimator-based* methods. Residual-based approaches (Yang et al., 2024a; Souček et al., 2025b) extract the watermark signal from one or a few images and transfer it directly onto an arbitrary target; while data-efficient, they rely on a high degree of content-agnosticity in the watermark and generalize poorly to content-dependent schemes (Souček et al., 2025b). Estimator-based approaches (Dong et al., 2025) instead train a model on a collection of watermarked images to learn the underlying watermark distribution, offering broader generalization. We focus on the latter paradigm and define the following threat model.

Threat Model: The adversary aims to generate a forged watermarked image that visually resembles a given clean image while being detected by the target watermarking scheme as containing a specified target message. However, the adversary has no knowledge of the internal watermarking algorithm and no access to the detection API. Instead, the adversary may only collect a limited subset of watermarked images distributed by the service provider and must rely solely on this data to infer patterns that enable successful forgery.

Inspired by recent advances in watermark removal attacks (Zhao et al., 2024; Liu et al., 2025; Hu et al., 2025), we investigate whether these techniques can be repurposed to approximate the distribution of non-watermarked images and subsequently enable watermark forgery. Specifically, we leverage watermark removal methods to construct a proxy dataset that approximates original non-watermark images from watermarked samples. Using this proxy representation, we train a conditional diffusion model (Ho & Salimans, 2021) capable of generating watermark signals conditioned on this proxy representation. Latent-refinement is applied after both the removal and forgery stages to improve visual fidelity.

Empirically, we observe that this mechanism results in a more data-efficient and robust forgery approach. In particular, our proposed approach—referred to as **EraseForge**—is able to circumvent previously proposed defense mechanisms, such as *multi-message strategy* (Dong et al., 2025). These findings indicate that existing defenses may remain vulnerable to forgery pipelines that exploit watermark removal and generative reconstruction. To mitigate this threat, we introduce a self-verifiable message generation mechanism comprising two complementary variants. The first, verifiable pseudo-random message, enforces internal message consistency and is effective against estimator-based forgery approaches including EraseForge. The second, content-dependent message, binds the watermark to the visual content of the host image, additionally neutralizing residual-based one-shot attacks (Souček et al., 2025b). Importantly, neither variant requires modifications to the core watermarking architecture, allowing existing pipelines to remain unchanged.

Our core contributions are as follows:

1. We introduce **EraseForge**, a diffusion-based forgery attack that utilizes watermark removal attacks to formulate an effective watermark forgery attack.
2. Through extensive experiments, we demonstrate that **EraseForge** successfully launches forgery attacks against a wide range of watermarking schemes while achieving greater robustness and data efficiency than existing state-of-the-art forgery methods.
3. We propose a self-verifiable message defense mechanism with two complementary variants—verifiable pseudo-random message and content-dependent message—that collectively neutralize both estimator-based and residual-based forgery attacks.
4. We study the impact of different perceptual refinement strategies on watermark removal attacks and introduce a *latent-space perceptual refinement* method that improves perceptual quality while minimally affecting removal efficacy.

2 PRELIMINARY: DIFFUSION PROCESS

Diffusion Models define a two-phase generative framework: a fixed forward process that incrementally corrupts data with Gaussian noise, and a learned reverse process that progressively denoises latent variables back into data samples (Ho et al., 2020; Sohl-Dickstein et al., 2015). Let $x_0 \sim q(x)$ denote a sample from the data distribution. The forward process defines a Markov chain that gradually adds Gaussian noise to x_0 over T time steps according to a fixed variance schedule $\{\beta\}_{t=1}^{T}$ with $\beta \in (0, 1)$. We directly generate x_t from x_0 using the reparameterization trick: $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ is the standard Gaussian noise and $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ is the cumulative noise schedule. A neural network ϵ_θ is trained to predict the noise added to x_0 at step t by minimizing:

$$\mathbb{E}_{t\sim[1,T],x_0\sim q(x),\ \epsilon\sim\mathcal{N}(0,I)}\left[\|\epsilon_\theta(x_t,t)-\epsilon\|_2^2\right]. \tag{1}$$

Denoising Diffusion Implicit Models (DDIM) (Song et al., 2021) provide a deterministic sampling formulation for diffusion models, enabling faster generation and trajectory inversion. Starting from $x_T \sim \mathcal{N}(0, I)$, the sample is iteratively denoised according to,

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(x_t,t)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}}\,\epsilon_\theta(x_t,t), \tag{2}$$

Unconditional diffusion models can be readily extended to conditional generation (Ho & Salimans, 2021; Rombach et al., 2022), by supplying auxiliary information c to the denoising network, where the learning objective becomes:

$$\mathbb{E}_{t\sim[1,T],x_0\sim q(x),\ \epsilon\sim\mathcal{N}(0,I)}\left[\|\epsilon-\epsilon_\theta(x_t,t,c)\|^2\right]. \tag{3}$$

Conditioning information is typically injected through feature concatenation, encoder features, or cross-attention mechanisms within the denoising network.

3 ERASEFORGE

In this section we introduce the **EraseForge** pipeline, which consists of 4 stages: (1) Watermark removal (2) Post removal refinement (3) Forgery Estimation and (4) Post forgery refinement. In abstract, we utilize watermark removal and post removal refinement to construct the training dataset containing watermark removed images and estimate the corresponding watermark residuals.

3.1 WATERMARK REMOVAL

We follow the watermark removal approach introduced by Zhao et al. (2024) (Zhao et al., 2024). Given a watermarked image x_w, we seek to approximate the watermark-removed image $\hat{x}_r$ that suppresses the watermark signal while preserving the semantic content. Following Zhao et al. (2024), we employ a diffusion model consisting of a variational autoencoder (VAE) encoder $\mathcal{V}_{enc}$ and decoder $\mathcal{V}_{dec}$ to formulate the latent representation $z_0 = \mathcal{V}_{enc}(x_w)$. Subsequently, a partial forward diffusion process is applied to z_0 to introduce Gaussian noise up to a specific timestep τ. The corrupted latent z_τ is defined by the linear combination:

$$z_\tau = \sqrt{\bar{\alpha}_\tau}z_0 + \sqrt{1-\bar{\alpha}_\tau}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{4}$$

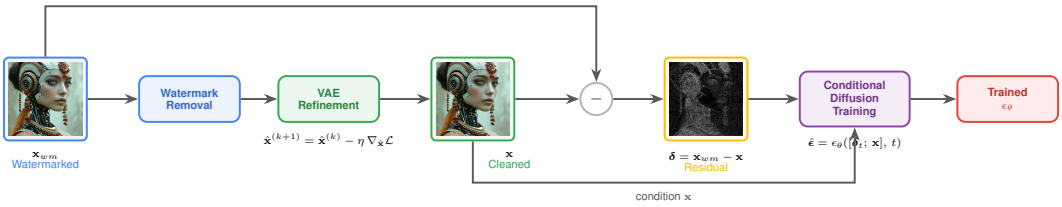
The recovery of the image is performed by a reverse diffusion process where the denoised latent $\hat{z}_0$ is passed through the VAE decoder: i.e., $\hat{x}_r = \mathcal{V}_{dec}(\hat{z}_0)$.

3.2 POST REMOVAL REFINEMENT

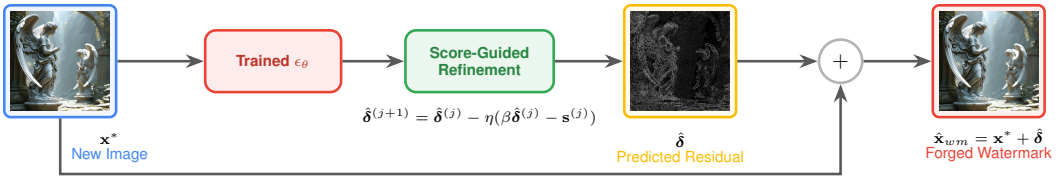
The regenerated image $\hat{x}_r$ may occasionally exhibit reconstruction artifacts, such as color shifts or blurring, stemming from the diffusion regeneration process. To mitigate these, we refine the output by optimizing in the VAE latent space to minimize a combined pixel and perceptual loss against the original watermarked reference x_w.

Let $z_r = \mathcal{V}_{enc}(\hat{x}_r)$ be the initial latent of the reconstructed image. We optimize z_r over N_r gradient steps:

$$z_r^{(n+1)} = z_r^{(n)} - \lambda\nabla_{z_r}(\lambda_1\left\|\mathcal{V}_{dec}\left(z_r\right) - x_w\right\|_2^2 + \lambda_2\left\|\phi\left(\mathcal{V}_{dec}\left(z_r\right)\right) - \phi(x_w)\right\|_2^2) \tag{5}$$



(a) Training



(b) Sampling

Figure 1: Overview of the proposed watermark forgery method. (a) Training pipeline for EraseForge approach where we first remove the watermark from the watermarked image and conduct refinement to improve perceptual quality. Then the removed and refined image is used to estimate the watermark residual which is then used to train a conditional diffusion model. (b) Sampling approach where an attacker use EraseForge to embed a forged watermark on arbitrary image

Here $\phi(\cdot)$ denotes VGG-19 feature extraction, λ_1 weights the pixel reconstruction term, λ_2 weights the perceptual term and λ controls the update strength.

After optimization, a global luminance correction is applied: $x_r = x_r \cdot \frac{\mu(x_w)}{\mu(x_r)}$ where $\mu(\cdot)$ calculates the mean pixel intensity to ensure photometric consistency and x_r denotes the refined image.

As an alternative to the aforementioned *latent-space perceptual refinement*, one may also consider a *score-guided Langevin correction* mechanism. In this framework, the refinement step is computed iteratively as:

$$x_r^{(n+1)} = x_r^n + \eta_r \nabla_{x_r^n} \left[\log p_r(x_r^n) - \lambda_r \|x_r^n - x_w\|^2 \right] \tag{6}$$

where η_r represents the step size or learning rate of the update and λ_r is a regularization weight balancing the prior and the fidelity term. The $\log p_r(\cdot)$ constrains the sample to reside in the watermark-removed image space, while the Mean Squared Error (MSE) term $\|x_r^n - x_w\|^2$ ensures the refined image remains perceptually consistent with the reference x_w.

3.3 Forgery Estimation

We train a conditional diffusion model to generate the estimated watermark residual $\delta = x_w - x_r$ rather than the full watermark image x_w. This residual formulation simplifies the learning task by restricting the model to structured perturbations introduced by the watermark. Instead of modeling the entire image distribution, the model trains solely on the watermark-induced modifications.

We follow stages 3.1 and 3.2 to construct the dataset $\mathcal{D} = \{x_r^{(i)}, \delta^{(i)}\}_{i=1}^N$. Many watermarking schemes, particularly content-dependent ones, exhibit localized spatial structure. By conditioning on x_r, the model can capture such localized patterns during learning. Consequently, the denoising network ϵ_θ is a conditional diffusion model whose forward process is captured by

$$\delta_t = \sqrt{\bar{\alpha}_t}\delta_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{7}$$

where δ_0 denotes the watermark residual and $\bar{\alpha}_t$ follows the standard cumulative noise schedule. The network is trained using the standard noise-prediction objective in Eq 3:

$$\mathbb{E}_{t\sim[1,T],\delta_0\sim q(\delta),\ \epsilon\sim\mathcal{N}(0,I)} \left[\|\epsilon - \epsilon_\theta(\delta_t, t, x_r)\|^2 \right]. \tag{8}$$

During inference, the objective is to generate a forged image x_f by integrating a structured perturbation into the original input x. This process utilizes a DDIM sampling strategy to estimate the watermark residual δ and compute $x_f = x + \delta$. Modeling the watermark residual instead of directly

regenerating the image helps preserve the original image structure and reduces artifacts that may arise from full diffusion-based image reconstruction.

3.4 Post Forgery Refinement

To further enhance perceptual quality, we incorporate a post-forgery refinement phase utilizing the score-guided Langevin correction. The iterative update is formulated as:

$$\delta^{(n+1)} = \delta^{(n)} + \eta_d \nabla_{\delta^{(n)}} \left[\log p_\delta(\delta^{(n)}) - \lambda_d (\delta^{(n)})^2 \right] = (1 - 2\eta_d \lambda_d)\delta^{(n)} + \eta_d \nabla_{\delta^{(n)}} \log p_\delta(\delta^{(n)}) \quad (9)$$

In this formulation, η_d denotes the step size of the update, while λ_d serves as a regularization weight that balances the learned prior against the fidelity term. Mechanistically, the score term $\nabla_{\delta^{(n)}} \log p_\delta(\cdot)$ steers the sample toward the high-density manifolds of the learned watermark residual distribution. Concurrently, the term $-2\eta_d \lambda_d \delta^{(n)}$ exerts a restoring force toward the origin, ensuring the perturbation remains bounded and perceptually unobtrusive.

4 EraseForge: Results and Discussion

4.1 Experimental Setup

We train a conditional diffusion model with a UNet backbone to model the watermark residual, conditioned on the watermark-removed and refined image x_r. For the diffusion model, we adopt DDIM sampling with total steps = 100 and use post-forgery refinement with $\lambda_d = 50$ and refinement iterations $L = 100$. A more detailed description about the training and watermarking setup is provided in Appendix A. We evaluate EraseForge across four watermarking schemes: three post-hoc methods—, Hidden(Zhu et al., 2018), RivaGAN(Zhang et al., 2019) and DWT-DCT(Al-Haj, 2007)—and one in-generation method, Stable Signature(Fernandez et al., 2023). Following the experimental protocol of WMCopier(Dong et al., 2025), we train on AI-generated images from DiffusionDB(Wang et al., 2022) and evaluate on both AI-generated and real photographs, using MS-COCO(Lin et al., 2014), ImageNet(Russakovsky et al., 2015), and CelebA-HQ(Karras et al., 2018) as real-world benchmarks while using DiffusionDB for AI-generated images. We consider two metrics: Peak Signal-to-Noise Ratio (PSNR) to evaluate visual quality and bit-accuracy to evaluate performance of the forgery approach. We detail the experimental setup and training configuration in Appendix A in more detail.

4.2 Comparative Performance

As illustrated in Table 1, EraseForge demonstrates substantially greater data efficiency relative to prior work. Notably, EraseForge achieves performance comparable to that of the original watermarking scheme while utilizing only 1,000 training samples, representing a $5\times$ improvement in data efficiency over existing approaches. Given that this reduction in data requirements translates directly to reduced computational and financial cost for an adversary mounting a forgery attack, EraseForge presents a considerably elevated threat to watermarking service providers.

We attribute this efficiency gain to two key design choices: residual modeling and conditional modeling. Together, these components simplify the forgery estimation objective by constraining the model to learn solely the underlying watermarking pattern, rather than requiring it to jointly model both the watermarking pattern and the full image distribution—as is necessary in unconditional approaches. We direct the reader to Appendix B for a comprehensive ablation study comparing conditional residual modeling against conditional watermark modeling. As reported in Table 7, while conditional watermark modeling yields lower bit accuracy relative to residual modeling, it nonetheless achieves meaningfully superior performance compared to existing forgery baselines. However, this approach introduces perceptible visual artifacts, an inherent consequence of diffusion-based generation. Residual modeling circumvents this limitation by restricting generation to the residual signal, thereby avoiding the synthesis of full image content and the associated visual degradation.

We further observe comparatively reduced performance under the DWT-DCT watermarking scheme, which we attribute to the inherent limitations of that approach—specifically, DWT-DCT achieves lower genuine bit accuracy on evaluation images relative to the neural watermarking methods, imposing a lower ceiling on forgery performance. With respect to visual fidelity, EraseForge achieves results comparable to those reported in prior literature (Dong et al., 2025; Müller et al., 2024).

Table 1: Watermark forgery performance (PSNR / Bit-accuracy) across attack methods and training dataset sizes. cells indicate bit-accuracy values above 90%.

		Wang et al. (2021)(Wang et al., 2021) (Blackbox)		Yang et al. (2024)(Yang et al., 2024a) (No box)		WMCopier (No box)				EraseForge (No box)			
						$N = 5000$		$N = 1000$		$N = 5000$		$N = 1000$	
Scheme	**Dataset**	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc
Hidden	MS-COCO	31.02	80.56	29.68	63.12	31.74	99.34	31.37	88.61	30.63	99.11	30.92	99.22
	CelebAHQ	31.57	82.28	29.79	61.52	33.12	98.08	33.50	87.31	33.98	99.39	34.42	99.37
	ImageNet	31.24	78.61	29.78	62.66	31.76	98.99	32.46	87.37	30.63	99.03	30.91	98.98
	DiffusionDB	30.74	79.99	29.68	63.36	31.46	98.83	33.28	88.57	31.63	99.18	31.78	99.21
RivaGAN	MS-COCO	32.94	93.26	29.12	50.80	34.07	95.74	34.16	81.82	31.09	99.59	31.21	97.13
	CelebAHQ	32.64	93.67	29.23	52.29	35.28	98.61	35.74	89.10	34.69	99.91	31.09	99.45
	ImageNet	33.11	90.94	29.22	50.92	33.87	93.83	34.10	81.25	30.96	99.13	26.93	97.71
	DiffusionDB	33.31	89.69	29.12	48.70	34.50	90.43	34.77	74.76	32.57	95.77	28.15	92.53
Stable Signature	MS-COCO	28.87	91.68	30.77	52.67	31.29	98.04	31.89	86.69	30.04	99.61	29.56	99.65
	CelebAHQ	32.33	79.90	30.51	51.73	30.54	96.04	32.89	85.56	33.23	99.18	33.28	99.09
	ImageNet	29.59	85.77	30.75	51.59	31.33	97.03	32.12	85.18	29.10	99.52	29.56	99.47
	DiffusionDB	31.11	89.24	30.65	52.69	31.59	96.24	31.61	84.30	29.80	99.03	30.33	98.99
DWT-DCT	MS-COCO	31.33	74.32	32.87	53.08	33.69	89.19	32.18	64.87	31.03	91.32	30.90	90.74
	CelebAHQ	32.19	81.29	32.90	53.68	35.29	89.46	32.51	64.82	34.52	90.07	34.57	88.98
	ImageNet	30.16	79.64	32.92	51.96	33.75	88.25	32.57	64.64	30.97	88.11	30.86	88.09
	DiffusionDB	31.87	78.22	32.90	51.59	33.84	85.17	33.04	67.66	32.06	87.82	31.89	87.58

Table 2: Bit accuracy (%) under various distortions. Results compare Genuine watermarks, WMCopier (5000 training samples), and EraseForge (1000 training samples). cells indicate the value is more than 10% below Genuine; cells indicate more than 20% below Genuine.

		JPEG			Blur			Gaussian Noise			Brightness		
Watermark Scheme	**Dataset**	Genuine	WMCopier (5000)	EraseForge (1000)	Genuine	WMCopier (5000)	EraseForge (1000)	Genuine	WMCopier (5000)	EraseForge (1000)	Genuine	WMCopier (5000)	EraseForge (1000)
Hidden	MS-COCO	58.68	58.06	59.81	78.50	71.95	73.30	54.13	49.55	60.89	82.40	78.99	81.47
	CelebAHQ	57.07	55.07	56.08	79.83	69.07	70.38	48.94	46.02	54.59	83.63	73.21	80.21
	ImageNet	58.86	57.83	58.42	78.20	71.34	72.40	54.10	49.57	60.95	80.95	77.40	82.82
	DiffusionDB	58.57	57.61	58.79	79.69	72.89	71.92	54.41	50.19	60.64	81.53	77.66	81.83
RivaGAN	MS-COCO	99.44	99.32	95.24	99.60	94.99	98.14	85.71	75.00	96.50	84.51	78.81	77.90
	CelebAHQ	99.92	97.22	98.90	99.97	98.23	99.35	85.93	74.83	98.24	85.60	79.53	80.11
	ImageNet	98.95	92.00	96.43	99.28	93.89	97.40	84.95	74.74	95.41	82.77	77.25	75.90
	DiffusionDB	96.56	84.85	90.41	97.27	86.96	92.01	77.33	66.27	89.99	79.14	71.65	73.58
Stable Signature	MS-COCO	93.99	89.48	93.76	86.91	68.34	92.21	73.48	67.14	94.50	92.30	88.63	78.09
	CelebAHQ		86.73	91.56		65.42	83.41		65.33	86.00		86.86	82.08
	ImageNet		87.73	93.41		64.88	91.60		61.79	93.84		91.41	75.39
	DiffusionDB		85.69	92.76		65.45	88.80		61.60	92.36		87.45	79.35
DWT-DCT	MS-COCO	56.44	53.00	52.71	59.84	56.56	53.34	67.86	66.90	83.30	54.66	58.36	56.29
	CelebAHQ	55.42	53.14	53.71	63.12	58.26	58.23	64.84	66.49	81.26	53.89	57.73	53.44
	ImageNet	56.08	52.31	52.65	59.37	54.39	52.82	68.27	67.60	82.12	54.08	57.37	53.82
	DiffusionDB	58.16	53.23	52.63	62.12	55.74	56.76	66.90	64.43	79.94	54.73	56.83	54.63

4.3 Robustness

To investigate the robustness of forged watermarks under real-world perturbations, we evaluate bit accuracy of both genuine and forged watermarked images under four common image distortions following the experimental setup of WMCopier (Dong et al., 2025): JPEG compression, Gaussian blur, Gaussian noise, and brightness adjustment. As Stable Signature does not support watermark embedding into arbitrary images, results for this scheme are reported exclusively on model-generated images. As shown in Table 2, forged watermarks produced by EraseForge achieve bit accuracy broadly comparable to, and in several instances exceeding, the genuine watermark baseline—while requiring only 1,000 training samples, compared to the 5,000 samples used by WMCopier. Across the Hidden and Stable Signature schemes, our method consistently matches or surpasses genuine bit accuracy across all distortion types and datasets. Notably, the most pronounced gains are observed under Gaussian noise, which we attribute to an inductive bias of the diffusion model coupled with residual modeling toward learning smooth, structured residuals, enabling more effective separation of the watermark signal from high-frequency stochastic perturbations. Overall, EraseForge exhibits substantially fewer failure cases than WMCopier—no instance of degradation exceeding 20 percentage points relative to the genuine baseline, compared to at least three such cases for WMCopier—demonstrating more stable forgeries across diverse schemes and distortions at a $5\times$ reduction in training data.

We conduct additional ablation studies in Appendix B to further characterise EraseForge's design choices, including: (i) the effect of watermark removal strength on forgery performance via a controlled blending mechanism, (ii) a comparison of residual modeling versus direct watermark modeling as the forgery objective, (iii) an evaluation of latent-space perceptual refinement versus score-guided Langevin correction, and (iv) a sensitivity analysis of the post-forgery refinement hyperparameters.

5 Defense: Self-Verifiable Message Generation

Based on the empirical finding of prior work that increasing the number of messages increases the difficulty of forgery estimation, we conducted a similar experiment and observed consistent results (see Table 9 in Appendix C). As the number of messages increases, the forgery estimator must

Table 3: Forgery performance of EraseForge against the multi-message defense (K=100) across varying training-set sizes. cells indicate bit-accuracy values above 90% indicating successful forgery.

		$N = 1000$				$N = 5000$				$N = 20000$			
		WMCopier		EraseForge		WMCopier		EraseForge		WMCopier		EraseForge	
Defense	**Dataset**	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc
100 messages	MS-COCO	34.63	68.96	28.01	69.77	34.86	71.56	28.72	72.60	34.78	71.91	30.26	92.73
	CelebAHQ	35.65	68.86	34.86	69.70	35.87	72.91	32.41	72.94	34.15	72.97	33.96	92.39
	ImageNet	34.82	69.10	32.34	69.56	34.58	71.44	28.64	72.73	34.57	72.56	30.21	91.87
	DiffusionDB	34.50	68.80	32.82	69.21	35.40	72.28	30.01	72.51	34.99	72.34	31.63	91.87

generalize across a larger and more diverse message space, rendering the estimation task inherently more challenging. To counter this, prior work (Dong et al., 2025) introduced a *multi-message strategy* where the message is randomly selected from a set of pre-defined messages. During detection, the verifier confirms the presence of any valid message within the pool, rather than a single fixed target. WMCopier reported that under this defense with K = 100, their forgery approach failed to learn a reliable forgery even when trained on 50,000 samples, suggesting that sufficient message diversity effectively neutralizes discriminative forgery estimators that rely on direct watermark pattern matching. As reported in Table 3, EraseForge demonstrates a markedly different scaling behavior under this defense. Crucially, EraseForge achieves over 91% bit accuracy at N = 20,000—well below the 50,000—sample threshold at which WMCopier was reported to fail—demonstrating that our residual modeling framework poses a credible and data-efficient threat to the multi-message defense mechanism, and raising important questions about the long-term viability of the multi-message defense strategy against estimator-based forgery attacks.

5.1 Self-verifiable Message

Although one can continue to increase the number of messages, this is not a scalable solution, as it requires storing multiple messages and increases the probability of a random message colliding with the predefined set. To address this, we propose the *self-verifiable message* mechanism. The core idea is to replace a finite message codebook with a deterministic relationship between the embedded message and a verifiable quantity, so that verification becomes a structural consistency check rather than a lookup over a finite pool. We present two complementary instantiations of this idea, which we refer to as **SeedBind** and **ContentBind**.

SeedBind: Verifiable pseudo-random message. Given a watermarking scheme that encodes an M-bit message, we randomly generate a k-bit seed $\hat{m}$ and employ a pseudo-random generation function $g(\cdot)$ that takes $\hat{m}$ as input to deterministically produce the remaining $M - k$ bits. The final embedded message is thus constructed as: $\mathbf{m} = [\hat{m} \ ; \ g(\hat{m})]$ where $[\cdot \ ; \ \cdot]$ denotes concatenation. During verification, the detector extracts the decoded message, recovers the seed $\hat{m}$ from the first k bits, recomputes $g(\hat{m})$, and verifies consistency with the remaining decoded bits. This formulation transforms watermark verification from a lookup problem over a finite message pool into a structural consistency check. In our evaluation we instantiate $g(\cdot)$ using a Locality-Sensitive Hashing (LSH) function based on the Simhash approach (Charikar, 2002) (details in Appendix D). We note, however, that the framework is agnostic to the specific choice of $g(\cdot)$: cryptographic hash functions (Krawczyk et al., 1997), other LSH schemes, or Linear Feedback Shift Registers (LFSRs) (Golomb, 2017) are all valid alternatives depending on the desired security–fidelity trade-off. A critical practical consideration is the tolerance of the verification function to bit-level decoding errors. The chosen function must either exhibit inherent robustness to bit-flips—as with LSH-based constructions that tolerate approximate matching—or be complemented by error-correcting codes (ECC) (Lin & Jr., 1983).

Table 4 compares the effectiveness of the multi-message defense with $K = 100$ against the verifiable pseudo-random message-based SeedBind defense across varying training-data regimes ($N \in \{1000, 5000, 20000\}$), evaluated on RivaGAN. Here $k = 16$ bits are sampled uniformly at random and $g(\cdot)$ deterministically generates the remaining bits, yielding a full 32-bit self-verifiable message. The SeedBind reduces forged bit accuracy to near-random levels ($\sim$52–59%) even at $N = 20{,}000$, where the 100-message pool allows >92%.

Limitation: single image residual-based forgery attacks. Although SeedBind is effective against forgery estimator training, it is still vulnerable to single image residual-based forgery attacks such

Table 4: Effectiveness of self-verifiable message defenses against EraseForge on RivaGAN. cells indicate bit-accuracy values above 90% indicating unsuccessful defense.

	$N = 1000$				$N = 5000$				$N = 20000$			
	100 messages		SeedBind		100 messages		SeedBind		100 messages		SeedBind	
Dataset	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc
MS-COCO	28.01	69.77	25.51	52.65	28.72	72.60	25.23	52.14	30.26	92.73	26.89	58.92
CelebAHQ	34.86	69.70	29.84	53.50	32.41	72.94	29.53	52.13	33.96	92.39	24.62	58.42
ImageNet	32.34	69.56	25.41	53.27	28.64	72.73	25.20	53.38	30.21	91.87	26.73	59.06
DiffusionDB	32.82	69.21	26.75	53.70	30.01	72.51	26.51	52.78	31.63	91.87	28.44	58.63

Table 5: Forged bit accuracy (%) under random, SeedBind and ContentBind. Lower is better for the defender; ∼50% indicates near-random—highlighted in –(forgery neutralized).

	CIN			**MBRS**			**Videoseal**		
Dataset	Random	SeedBind	ContentBind	Random	SeedBind	ContentBind	Random	SeedBind	ContentBind
MS-COCO	100.0	100.0	51.0	84.9	70.6	50.5	86.5	74.5	50.5
CelebAHQ	100.0	100.0	51.4	90.4	74.6	52.0	88.5	74.7	52.1
ImageNet	100.0	100.0	51.1	84.3	70.3	50.7	86.0	73.0	50.9
DiffusionDB	99.9	99.4	52.5	84.6	70.8	52.7	82.3	72.6	52.2

as WMForger (Souček et al., 2025b). WMForger estimates the watermark residual from a *single* watermarked image and transfers it directly onto an unwatermarked target. Because the residual is extracted from one specific image, it carries the embedded message—including the seed $\hat{m}$ and the correctly expanded bits $g(\hat{m})$. When this residual is pasted onto the target, the forged image inherits a structurally valid message that passes the SeedBind's consistency check. In this setting, for vulnerable watermarking approaches, message randomisation offers no protection. This limitation motivates a second variant that binds the message not to an internal seed but to the *visual content* of the image itself.

ContentBind: Content-dependent Message Generation. To defend against both estimator-based and residual-based forgery attacks, we propose an alternative instantiation that binds the watermark message directly to the visual content of the host image, eliminating the need for a separate seed altogether. Given a feature extractor $\phi(\cdot)$ (e.g. a pretrained vision backbone such as DINOv2 (Oquab et al., 2024), CLIP (Radford et al., 2021), or ResNet (He et al., 2016)) and a binarisation function $h(\cdot)$, the embedded message is computed as: $\mathbf{m} = h\big(\phi(\mathbf{x})\big)$ where $\mathbf{x}$ is the original image. Verification proceeds by recomputing $h(\phi(\mathbf{x}'))$ on the received image $\mathbf{x}'$ and checking consistency with the decoded watermark message. We use DINOv2 as the feature extractor $\phi(\cdot)$ and outline the approach in more detail in Appendix D.

This design neutralizes residual-based attacks by construction. Even if the attacker perfectly extracts and transfers the watermark residual from a source image $\mathbf{x}_{\text{src}}$, the forged target $\mathbf{x}_{\text{tgt}}$ has different visual content. The verifier recomputes $h(\phi(\mathbf{x}_{\text{tgt}}))$, which diverges from the decoded message $h(\phi(\mathbf{x}_{\text{src}}))$ embedded via the source, causing verification to fail. Crucially, the attacker cannot circumvent this without altering the visual content of the forged image to match the source—which would defeat the purpose of the forgery.

We evaluate SeedBind and ContentBind against WMForger (Souček et al., 2025b) on the three watermarking schemes—CIN (Ma et al., 2022), MBRS (Jia et al., 2021), and VideoSeal (Fernandez et al., 2024)—against which WMForger reported the highest forgery success, providing a stringent test of our defense. Table 5 reports the forged bit accuracy under SeedBind and ContentBind across these schemes and four datasets. Under random (non-bound) messages, WMForger achieves 82–100% bit accuracy. With SeedBind, performance varies between 70–100%; notably, in cases where one-shot residual transfer succeeds perfectly (e.g., CIN), SeedBind offers no protection, consistent with the limitation discussed above. In contrast, ContentBind reduces accuracy to ∼50–53% across all encoders and datasets, confirming that the forged watermark is no longer verifiable regardless of the attack strategy employed.

We further acknowledge that forgery estimation can be made considerably more difficult by substituting the specific LSH or binarization functions considered here with stronger cryptographic hashing and by incorporating ECC to explicitly safeguard the embedded message against bit-flip errors arising from image distortions or adversarial post-processing. However, this may require watermarking schemes that support a larger payload than those considered in this study.

6 Related Work

Image Watermarking: Watermarking techniques offer a reliable and active mechanism for tracing the provenance of digital images and enforcing copyright protection (Gowal et al., 2025; Fernandez et al., 2024). We broadly categorize existing approaches into *post-hoc* and *in-generation* methods based on the stage at which the watermark is embedded. Early works in this category employ hand-crafted frequency-domain techniques, such as DCT and DWT-based transforms (Al-Haj, 2007; Navas et al., 2008; Bors & Pitas, 1996), which suffer from limited robustness. Neural watermarking approaches address these limitations by jointly optimizing encoder-decoder architectures end-to-end with an augmentation module that simulates common distortions during training and in certain cases employ GAN-based methods to further improve visual fidelity (Fernandez et al., 2024; Zhu et al., 2018; Zhang et al., 2019; Souček et al., 2025a; Tancik et al., 2020). In contrast, in-generation methods introduce watermarks during the image generation process by modifying components of the generative model (Huang et al., 2024). In-generation approaches include methods that fine-tune the VAE decoder to imperceptibly embed a fixed watermark into all generated images (Fernandez et al., 2023), as well as semantic watermarking methods that embed messages directly into the initial noise latent of the diffusion sampling process without modifying model weights (Wen et al., 2023; Yang et al., 2024b; Rebuffi et al., 2026).

Watermark Attacks: Attacks divide into *removal*—suppressing the watermark below the detection threshold—and *forgery*—implanting a target watermark to enable false attribution. Removal methods range from standard distortions (Fernandez et al., 2024) to diffusion-based regeneration (Zhao et al., 2024; Saberi et al., 2024; Rabbani et al., 2024), further improved by controlled regeneration (Liu et al., 2025), transfer attacks (Hu et al., 2025; Souček et al., 2025b), and adaptive optimization (Lukas et al., 2024). Forgery attacks pose a complementary and arguably more severe threat, as they enable false attribution of illicit content to Gen-AI providers. Early approaches framed forgery as image translation in the black-box or white box setting(Wang et al., 2021; Müller et al., 2024; Kutter et al., 2000; Kinakh et al., 2024), requiring query access to the watermarking scheme. More recent work has shifted toward no-box forgery(Yang et al., 2024a; Dong et al., 2025; Souček et al., 2025b), which we characterize into two categories: residual-based approaches which estimate the watermark residual from a collection of watermarked images and transfer it directly to a clean target(Yang et al., 2024a; Souček et al., 2025b) and forgery estimator training approaches(Dong et al., 2025), which learn the underlying watermark distribution from a collection of watermarked images. We focus specifically on the latter paradigm, owing to its demonstrated generalisability across diverse watermarking schemes.

Forgery Defense: Despite growing interest in forgery attacks, dedicated defenses remain sparse. The most direct countermeasure is the *multi-message strategy* (Dong et al., 2025), which replaces a single message with a pool of K random messages to degrade forgery accuracy; however, as we show, this scales poorly against generative pipelines. Orthogonal mitigations include asymmetric watermarking (Furon & Duhamel, 2003; Cox et al., 2007), fingerprinting codes (Tardos, 2008; Boneh & Shaw, 1998), and adversarial encoder training (Souček et al., 2025b). Our self-verifiable message mechanism differs fundamentally: rather than enlarging the message space or hardening the encoder, it transforms verification from a finite codebook lookup into a structural consistency check.

7 Conclusion

We presented EraseForge, a no-box watermark forgery attack that repurposes diffusion-based watermark removal to train a conditional residual diffusion model. By restricting generation to the watermark residual rather than the full image, EraseForge achieves state-of-the-art forgery performance with only 1,000 training samples and circumvents the multi-message defense at a fraction of the data cost where prior methods fail. These results expose a fundamental vulnerability in the multi-message defense approach against the EraseForge approach. To address this, we propose a self-verifiable message mechanism in two variants: SeedBind, which enforces internal message consistency against estimator-based attacks, and ContentBind, which ties the watermark to the image's visual content and additionally neutralizes residual-based one-shot attacks—reducing forged bit accuracy to $\sim$50% across all tested encoders without modifying the underlying watermarking architecture. We hope these findings motivate a shift toward cryptographically grounded verification in the design of provenance systems for generative media.

REFERENCES

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

Amazon. Watermark detection for amazon titan image generator now available in amazon bedrock. https://aws.amazon.com/cn/about-aws/whats-new/2024/04/watermark-detection-amazon-titan-image-generator-bedrock/, 2024.

Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data. In *IEEE Transactions on Information Theory*, volume 44, pp. 1897–1905, 1998.

A.G. Bors and I. Pitas. Image watermarking using DCT domain constraints. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pp. 231–234, Lausanne, Switzerland, 1996. IEEE. ISBN 978-0-7803-3259-1. doi: 10.1109/ICIP.1996.560426.

California State Legislature. SB-942 california AI transparency act. https://leginfo.legislature.ca.gov/faces/billNavClient.xhtml?bill_id=202320240SB942, 2024.

Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pp. 380–388, 2002.

Ingemar J Cox, Matthew L Miller, Jeffrey A Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2nd edition, 2007.

Google Deepmind. Synthid: Identifying ai-generated content with synthid. https://deepmind.google/technologies/synthid/, 2023.

Google Deepmind. Imagen 4: Our leading text-to-image model, engineered for creativity. https://deepmind.google/models/imagen/, 2025.

Ziping Dong, Chao Shuai, Zhongjie Ba, Peng Cheng, Zhan Qin, Qinglong Wang, and Kui Ren. WMCopier: Forging invisible watermarks on arbitrary images. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=TbQCWblZAZ.

European Parliament and Council of the European Union. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (AI act). https://eur-lex.europa.eu/eli/reg/2024/1689/oj, 2024.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22466–22477, 2023.

Pierre Fernandez, Hady Elsahar, I Zeki Yalniz, and Alexandre Mourachko. Video seal: Open and efficient video watermarking. *arXiv preprint arXiv:2412.09492*, 2024.

Teddy Furon and Pierre Duhamel. An asymmetric watermarking method. In *IEEE Transactions on Signal Processing*, volume 51, pp. 981–995, 2003.

Solomon W Golomb. *Shift register sequences: secure and limited-access code generators, efficiency code generators, prescribed property generators, mathematical models*. World Scientific, 2017.

Lixue Gong, Xiaoxia Hou, Fanshi Li, Liang Li, Xiaochen Lian, Fei Liu, Liyang Liu, Wei Liu, Wei Lu, Yichun Shi, Shiqi Sun, Yu Tian, Zhi Tian, Peng Wang, Xun Wang, Ye Wang, Guofeng Wu, Jie Wu, Xin Xia, Xuefeng Xiao, Linjie Yang, Zhonghua Zhai, Xinyu Zhang, Qi Zhang, Yuwei Zhang, Shijia Zhao, Jianchao Yang, and Weilin Huang. Seedream 2.0: A native chinese-english bilingual image generation foundation model, 2025. URL https://arxiv.org/abs/2503.07703.

Sven Gowal, Rudy Bunel, Florian Stimberg, David Stutz, Guillermo Ortiz-Jimenez, Christina Kouridi, Mel Vecerik, Jamie Hayes, Sylvestre-Alvise Rebuffi, Paul Bernard, et al. Synthid-image: Image watermarking at internet scale. *arXiv preprint arXiv:2510.09263*, 2025.

Fan Guo, Jiyu Kang, Qi Ming, Emily Davis, and Finn Carter. Vanishing watermarks: Diffusion-based image editing undermines robust invisible watermarking, 2026. URL https://arxiv.org/abs/2602.20680.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

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

Yuepeng Hu, Zhengyuan Jiang, Moyang Guo, and Neil Zhenqiang Gong. A transfer attack to image watermarks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=UchRjcf4z7.

Huayang Huang, Yu Wu, and Qian Wang. ROBIN: Robust and invisible watermarks for diffusion models with adversarial optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=RvoxlFvnlX.

Zhaoyang Jia, Han Fang, and Weiming Zhang. 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*, MM '21, pp. 41–49, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450386517. doi: 10.1145/3474085.3475324. URL https://doi.org/10.1145/3474085.3475324.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

Vitaliy Kinakh, Brian Pulfer, Yury Belousov, Pierre Fernandez, Teddy Furon, and Slava Voloshynovskiy. Evaluation of security of ml-based watermarking: Copy and removal attacks, 2024. URL https://arxiv.org/abs/2409.18211.

Hugo Krawczyk, Mihir Bellare, and Ran Canetti. Hmac: Keyed-hashing for message authentication. *RFC*, 2104:1–11, 1997. URL https://api.semanticscholar.org/CorpusID:33812837.

Martin Kutter, Sviatoslav V Voloshynovskiy, and Alexander Herrigel. Watermark copy attack. In *Security and Watermarking of Multimedia Contents II*, volume 3971, pp. 371–380. SPIE, 2000.

Shu Lin and Daniel J. Costello Jr. *Error control coding - fundamentals and applications.* Prentice Hall computer applications in electrical engineering series. Prentice Hall, 1983. ISBN 978-0-13-283796-5.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Yepeng Liu, Yiren Song, Hai Ci, Yu Zhang, Haofan Wang, Mike Zheng Shou, and Yuheng Bu. Image watermarks are removable using controllable regeneration from clean noise. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=mDKxlfraAn.

Nils Lukas, Abdulrahman Diaa, Lucas Fenaux, and Florian Kerschbaum. Leveraging optimization for adaptive attacks on image watermarks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=O9PArxKLe1.

Rui Ma, Mengxi Guo, Yi Hou, Fan Yang, Yuan Li, Huizhu Jia, and Xiaodong Xie. Towards Blind Watermarking: Combining Invertible and Non-invertible Mechanisms. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 1532–1542, Lisboa Portugal, October 2022. ACM. doi: 10.1145/3503161.3547950.

Andreas Müller, Denis Lukovnikov, Jonas Thietke, Asja Fischer, and Erwin Quiring. Black-box forgery attacks on semantic watermarks for diffusion models. *arXiv preprint arXiv:2412.03283*, 2024.

K. A. Navas, Mathews Cheriyan Ajay, M. Lekshmi, Tampy S. Archana, and M. Sasikumar. DWT-DCT-SVD based watermarking. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, pp. 271–274. IEEE, January 2008.

Yunyi Ni, Ziyu Yang, Ze Niu, Emily Davis, and Finn Carter. On the information-theoretic fragility of robust watermarking under diffusion editing, 2025. URL https://arxiv.org/abs/2511.10933.

OpenAI. Dalle3. https://openai.com/index/dall-e-3/.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Qian Qi, Jiangyun Tang, Jim Lee, Emily Davis, and Finn Carter. Editing away the evidence: Diffusion-based image manipulation and the failure modes of robust watermarking, 2026. URL https://arxiv.org/abs/2603.12949.

Tahseen Rabbani, Bang An, Mucong Ding, Aakriti Agrawal, Yuancheng Xu, Chenghao Deng, Sicheng Zhu, Abdirisak Mohamed, Yuxin Wen, Tom Goldstein, and Furong Huang. WAVES: Benchmarking the robustness of image watermarks. In *Privacy Regulation and Protection in Machine Learning*, 2024. URL https://openreview.net/forum?id=tUumjNa7TF.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Sylvestre-Alvise Rebuffi, Tuan Tran, Valeriu Lacatusu, Pierre Fernandez, Tomáš Souček, Nikola Jovanović, Tom Sander, Hady Elsahar, and Alexandre Mourachko. Learning to watermark in the latent space of generative models, 2026. URL https://arxiv.org/abs/2601.16140.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Mehrdad Saberi, Vinu Sankar Sadasivan, Keivan Rezaei, Aounon Kumar, Atoosa Chegini, Wenxiao Wang, and Soheil Feizi. Robustness of AI-image detectors: Fundamental limits and practical attacks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=dLoAdIKENc.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Tomáš Souček, Pierre Fernandez, Hady Elsahar, Sylvestre-Alvise Rebuffi, Valeriu Lacatusu, Tuan Tran, Tom Sander, and Alexandre Mourachko. Pixel seal: Adversarial-only training for invisible image and video watermarking. *arXiv preprint arXiv:2512.16874*, 2025a.

Tomáš Souček, Sylvestre-Alvise Rebuffi, Pierre Fernandez, Nikola Jovanović, Hady Elsahar, Valeriu Lacatusu, Tuan Tran, and Alexandre Mourachko. Transferable black-box one-shot forging of watermarks via image preference models. In *Advances in Neural Information Processing Systems*, 2025b.

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

Gábor Tardos. Optimal probabilistic fingerprint codes. *Journal of the ACM*, 55(2):1–24, 2008.

The White House. Executive order on the safe, secure, and trustworthy development and use of artificial intelligence. https://www.federalregister.gov/documents/2023/11/01/2023-24283/safe-secure-and-trustworthy-development-and-use-of-artificial-intelligence, 2023.

Ruowei Wang, Chenguo Lin, Qijun Zhao, and Feiyu Zhu. Watermark faker: towards forgery of digital image watermarking. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2021.

Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv preprint arXiv:2210.14896*, 2022.

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

Pei Yang, Hai Ci, Yiren Song, and Mike Zheng Shou. Can simple averaging defeat modern watermarks? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a. URL https://openreview.net/forum?id=X2G7LA7Av9.

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

Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust invisible video watermarking with attention. *arXiv preprint arXiv:1909.01285*, 2019.

Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative AI. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=7hy5fy2OC6.

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

A Experimental Setup: Further Details

A.1 Watermarking schemes

We evaluate EraseForge on four watermarking schemes spanning both classical signal processing and modern deep learning paradigms. Following the experimental protocol of WMCopier (Dong et al., 2025), all watermarking approaches operate at a resolution of 512×512. Below we describe each scheme in detail.

HiDDeN (Zhu et al., 2018). HiDDeN is an end-to-end trainable framework for data hiding that comprises four components: an encoder E_θ, a parameterless noise layer N, a decoder D_ϕ, and an adversarial discriminator A_γ. The encoder receives a cover image and a binary message and produces a visually indistinguishable encoded image. The noise layer sits between encoder and decoder during training, simulating real-world distortions (Gaussian blur, pixel-wise dropout, cropping, and JPEG compression via differentiable approximations) to encourage robustness. The decoder recovers the hidden message from the (possibly distorted) encoded image, and the discriminator is trained to distinguish cover from encoded images, providing an adversarial loss that improves visual quality. Training jointly minimizes an image distortion loss, a message reconstruction loss, and the adversarial loss. We use the *32-bit* message.

RivaGAN (Zhang et al., 2019). RivaGAN is an end-to-end deep learning architecture originally proposed for robust video watermarking. Its key contribution is a novel attention-based embedding mechanism that learns a probability distribution over the data dimensions for each pixel, allowing the model to selectively attend to different bits based on the image content. This replaces the naïve spatial repetition used in prior work and yields a more compact, content-adaptive data representation. The architecture is trained with two independent adversarial networks: a *critic* that evaluates visual quality, encouraging imperceptible watermarks, and an *adversary* that attempts to remove the watermark, promoting robustness. Noise layers (scaling, cropping, and differentiable compression) are applied between encoder and decoder during training. It encodes a *32-bit* message. We use the pre-trained model provided by the `invisible-watermark`[1] library.

Stable Signature (Fernandez et al., 2023). Unlike the above post-processing methods, Stable Signature is an in-generation watermarking scheme that roots a watermark directly into the weights of a latent diffusion model. The method fine-tunes the VAE decoder of a latent diffusion model, conditioned on a binary signature, so that every image decoded through it imperceptibly carries the signature. A separate pre-trained watermark extractor network recovers the hidden signature from any generated image, and a statistical test determines whether the image originates from the watermarked model. This design enables both detection (is this AI-generated?) and identification (which model produced it?), and the watermark persists even under aggressive image modifications. It encodes a *48-bit* message. We use the publicly available fine-tuned decoder weights and the pre-trained TorchScript message extractor (`dec_48b_whit`).

DWT-DCT (Al-Haj, 2007). Unlike the above approaches, DWT-DCT is a classical frequency-domain watermarking method that embeds a binary message by first decomposing the image into sub-bands using the Discrete Wavelet Transform (DWT), then applying the Discrete Cosine Transform (DCT) on selected sub-bands to modify the resulting frequency-domain coefficients. It encodes a *32-bit* message. We use the implementation from the `invisible-watermark` library with default parameters. As a non-neural method, DWT-DCT requires no learned weights and operates entirely on the CPU.

Following the protocol established by WMCopier (Dong et al., 2025), we apply a data filtering step specific to DWT-DCT. DWT-DCT suffers from low bit accuracy on certain images, which leads to unreliable watermark detection and verification. To reflect a more practical scenario, we assume that the service provider only returns images with high bit accuracy to users to ensure traceability. Accordingly, we select only images with 100% bit accuracy to construct the watermark image dataset for this scheme. This filtering is not required for the neural watermarking methods (HiDDeN, RivaGAN, Stable Signature), which achieve consistently high embedding fidelity across diverse image content. Under this setup, WMCopier reported that their forgery achieves even higher bit accuracy than the original DWT-DCT watermarking process itself. As shown in Table 1, EraseForge

[1] `https://github.com/ShieldMnt/invisible-watermark`

achieves comparable or better performance compared to WMCopier on this scheme while using $5\times$ fewer training samples.

The following three schemes are additionally used in the ContentBind defense evaluation (Table 5).

CIN (Ma et al., 2022). CIN (Combining Invertible and Non-invertible mechanisms) is a blind watermarking framework that decomposes the embedding process into two complementary pathways. An invertible branch, built on invertible neural network (INN) blocks, embeds the watermark message through a bijective mapping that preserves image fidelity, while a non-invertible branch processes the cover image through a standard convolutional encoder to enhance robustness against common distortions. The two branches are fused to produce the final watermarked image, and a CNN-based decoder recovers the hidden message. This dual-pathway design allows CIN to achieve a favorable trade-off between imperceptibility and robustness. We use a *32-bit* message with the publicly available pre-trained weights.

MBRS (Jia et al., 2021). MBRS (Mini-Batch of Real and Simulated JPEG compression) is a DNN-based watermarking method that specifically targets robustness against JPEG compression. Its key contribution is a training strategy that combines real JPEG compression (non-differentiable) with differentiable JPEG simulation within each mini-batch. By mixing real and simulated JPEG distortions during training, the encoder–decoder network learns to embed watermarks that withstand actual JPEG artifacts rather than relying solely on differentiable approximations. The architecture follows the standard encoder–noise layer–decoder paradigm, where the noise layer applies a randomly selected combination of real and simulated JPEG at varying quality factors. We use a *256-bit* message with the publicly available pre-trained model.

VideoSeal (Fernandez et al., 2024). VideoSeal is an open and efficient watermarking method designed for video but equally applicable to still images. It employs a lightweight encoder–decoder architecture trained with an adversarial discriminator and a suite of augmentations tailored for video pipelines, including temporal jitter, compression, and geometric transformations. The encoder embeds a binary message into frames via an additive residual, and the decoder extracts the message from individual frames independently. VideoSeal is designed for efficiency and scalability, operating at high throughput suitable for real-time applications. We use a *96-bit* message with the publicly available pre-trained model.

A.2 Training and Sampling Details

Model architecture. The forgery estimation model is a conditional UNet operating in pixel space, implemented using the `UNet2DModel` from the HuggingFace Diffusers library[2]. The model takes a 6-channel input formed by concatenating the noisy diffusion sample $x_t \in \mathbb{R}^{3\times H\times W}$ with the conditioning image (the watermark-removed image $\hat{x}_r$) along the channel dimension, and predicts the noise $\epsilon \in \mathbb{R}^{3\times H\times W}$. The architecture uses block output channels $(128, 128, 256, 256, 512, 512)$ with 2 layers per block, and includes attention at the bottleneck via `AttnDownBlock2D` and `AttnUpBlock2D`.

Training. We train using AdamW with a learning rate of 2×10^{-5} for 1000 epochs. The diffusion process uses $T = 1{,}000$ timesteps with a linear noise schedule, instantiated as a DDIM scheduler (Song et al., 2021) with ϵ-prediction. Images are resized to 512×512 and normalized to $[-1, 1]$. Classifier-free guidance is enabled via condition dropout with probability $p_{\text{drop}} = 0.1$. We maintain an exponential moving average (EMA) of model weights with a decay factor of 0.995. Training is performed on a single `p4d.24xlarge` instance ($8\times$ NVIDIA A100 40 GB GPUs) using distributed data parallel with a per-GPU batch size of 4 (effective batch size of 32), without mixed precision.

Sampling. At inference, we use the DDIM sampler with 100 denoising steps and a guidance scale of 1.5. Starting from Gaussian noise $x_T \sim \mathcal{N}(0, \mathbf{I})$, the model iteratively denoises conditioned on x to produce the estimated watermark residual $\hat{\delta}$, which is added to x to form the forged watermarked image.

[2] `https://github.com/huggingface/diffusers`

Post-forgery refinement. After the final denoising step, we apply score-guided latent refinement to improve visual fidelity while preserving watermark accuracy. Specifically, for $M = 100$ iterations, where $s_\theta(x, t)$ is the score estimate derived from the trained diffusion model, and regularisation weight $\lambda_d = 50$ controls the content preservation strength.

A.3 METRICS

We primarily evaluate forgery performance along two axes: visual fidelity and watermark accuracy.

Peak Signal-to-Noise Ratio (PSNR). PSNR measures the pixel-level distortion between the forged watermarked image and the original clean image. Higher PSNR indicates that the forgery introduces less visible degradation. It is defined as $\mathrm{PSNR}(x, x_f) = -10 \cdot \log_{10}(\mathrm{MSE}(x, x_f))$, where x is the original clean image, x_f is the forged watermarked image and $MSE(\cdot)$ is the mean squared error.

Bit Accuracy. Bit accuracy measures the fraction of correctly recovered watermark bits when the watermark decoder is applied to the forged image. It is computed as the proportion of matching bits between the extracted message and the target message. A bit accuracy close to 100% indicates that the forged image is indistinguishable from a genuinely watermarked image from the perspective of the watermark detector.

SSIM and MS-SSIM. In addition to PSNR, we report the Structural Similarity Index (SSIM) and its multi-scale variant (MS-SSIM) in selected experiments to provide a perceptual quality assessment. Unlike PSNR, which captures pixel-wise distortion, SSIM and MS-SSIM evaluate structural and perceptual similarity between images, accounting for luminance, contrast, and structure at one or multiple spatial scales respectively.

B ABLATION STUDIES

B.1 ABLATION STUDY: IMPACT OF WATERMARK REMOVAL

To systematically study the impact of watermark removal attack strength on forgery performance, we introduce a controlled blending mechanism that interpolates between the original watermarked image and its watermark-removed counterpart. We do so by adding the watermarked image x_w in a controlled manner to the watermark-removed image $\hat{x}_r$ to form the blended image $x_b = \alpha x_w + (1 - \alpha)\hat{x}_r$ where $\alpha \in [0, 1]$ is the blending factor. The diffusion model is then trained conditioned on x_b in place of x_r. As reported in Table 6, forgery performance degrades monotonically as α increases, consistent with the intuition that greater residual watermark signal in the conditioning input reduces the model's ability to accurately estimate and reproduce the target watermark pattern. We also note the gradual increment in the visual fidelity metric as α decrease. We attribute this to the diminishing influence of $\hat{x}_r$ in the conditioning signal, which carries visual artifacts introduced by the diffusion-based watermark removal process.

Table 6: Effect of blending factor α on PSNR and bit-accuracy for the Hidden watermark scheme. Lower α retains less of the original watermark signal, yielding higher bit-accuracy.

	$\alpha = 0.5$		$\alpha = 0.3$		$\alpha = 0.1$		$\alpha = 0$	
Dataset	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc
Training data	33.40	91.08	30.93	79.48	28.90	63.80	30.58	59.13
MS-COCO	36.53	70.74	33.71	81.15	30.76	95.08	30.92	99.22
CelebAHQ	39.23	67.55	36.80	74.94	33.70	92.59	34.42	99.37
ImageNet	36.25	69.74	33.47	79.43	30.41	94.17	30.91	98.98
DiffusionDB	36.50	70.06	33.89	79.56	31.23	93.76	31.78	99.21

B.2 ABLATION STUDY: RESIDUAL MODELING VS. DIRECT WATERMARK MODELING

A central design choice in EraseForge is to train the conditional diffusion model on the watermark residual $\delta = x_w - x_r$ rather than on the full watermarked image x_w. To validate this choice, we compare the two objectives under identical training conditions (1,000 samples, same UNet backbone and conditioning). As reported in Table 7, residual modeling consistently achieves higher PSNR across all watermarking schemes and datasets while maintaining comparable or superior bit accuracy. Direct watermark modeling attains competitive bit accuracy but incurs a substantial drop in visual

fidelity (approximately 8–10 dB lower PSNR), as the diffusion model must synthesize the full image content, introducing perceptible reconstruction artifacts. These results confirm that restricting generation to the residual signal simplifies the learning objective and preserves the original image structure.

Table 7: Residual modeling vs. watermark modeling: PSNR and bit-accuracy comparison across watermark schemes and datasets.

		Residual Modeling		Watermark Modeling	
Scheme	**Dataset**	PSNR	Bit-acc	PSNR	Bit-acc
Hidden	MS-COCO	30.92	99.22	22.05	97.18
	CelebAHQ	34.42	99.37	22.53	96.90
	ImageNet	30.91	98.98	20.90	97.31
	DiffusionDB	31.78	99.21	22.61	97.61
RivaGAN	MS-COCO	31.21	97.13	22.47	96.13
	CelebAHQ	31.09	99.45	23.02	97.38
	ImageNet	26.93	97.71	21.27	95.66
	DiffusionDB	28.15	92.53	23.14	90.33
Stable Signature	MS-COCO	29.56	99.65	23.85	98.62
	CelebAHQ	33.28	99.09	23.97	98.05
	ImageNet	29.56	99.47	22.64	98.41
	DiffusionDB	30.33	98.99	24.22	98.00
DWT-DCT	MS-COCO	30.90	90.74	25.59	77.40
	CelebAHQ	34.57	88.98	26.27	75.13
	ImageNet	30.86	88.09	24.62	76.37
	DiffusionDB	31.89	87.58	26.99	71.39

B.3 Ablation Study: Post-Forgery Refinement Hyperparameters

The post-forgery refinement stage (Section 3.4) is governed by two hyperparameters: the number of refinement steps and the regularization weight λ_d, which balances the learned score prior against the magnitude penalty. We analyze their effect on forgery quality across all four watermarking schemes using the MS-COCO dataset.

Figure 2 varies the number of refinement steps while fixing $\lambda_d = 100$. PSNR increases rapidly with additional steps and saturates around 100–150 steps for most schemes, while bit accuracy remains largely stable for Hidden, RivaGAN, and Stable Signature. DWT-DCT exhibits a more pronounced accuracy–fidelity trade-off, with bit accuracy declining steadily beyond 100 steps.

Figure 3 varies λ_d while fixing the number of refinement steps to 100. Increasing λ_d improves PSNR by encouraging smaller perturbations, at the cost of a gradual reduction in bit accuracy. These results motivate the default configuration of $\lambda_d = 50$ and 100 refinement steps used in our main experiments, which balances forgery fidelity and accuracy across schemes.

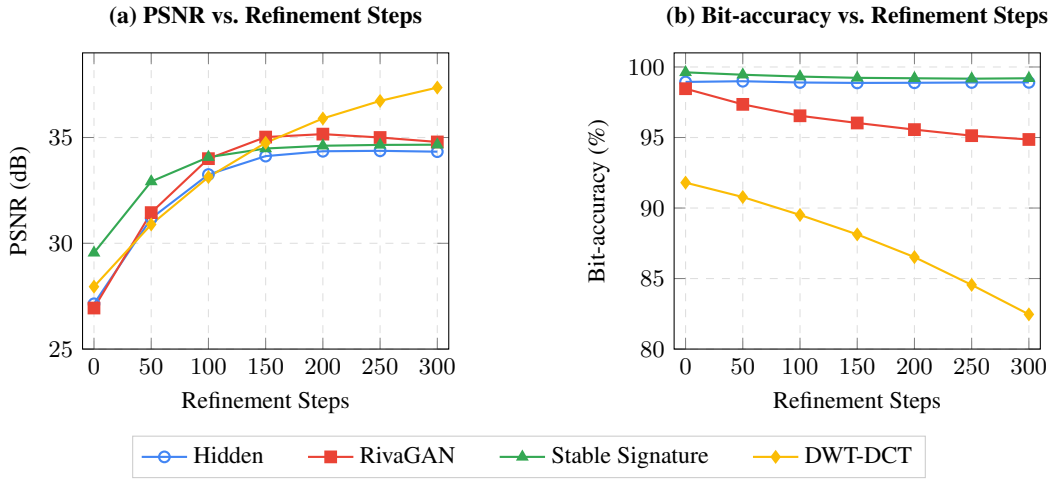


Figure 2: Effect of the number of refinement steps on PSNR and bit accuracy (regularization weight $\lambda_d = 100$).

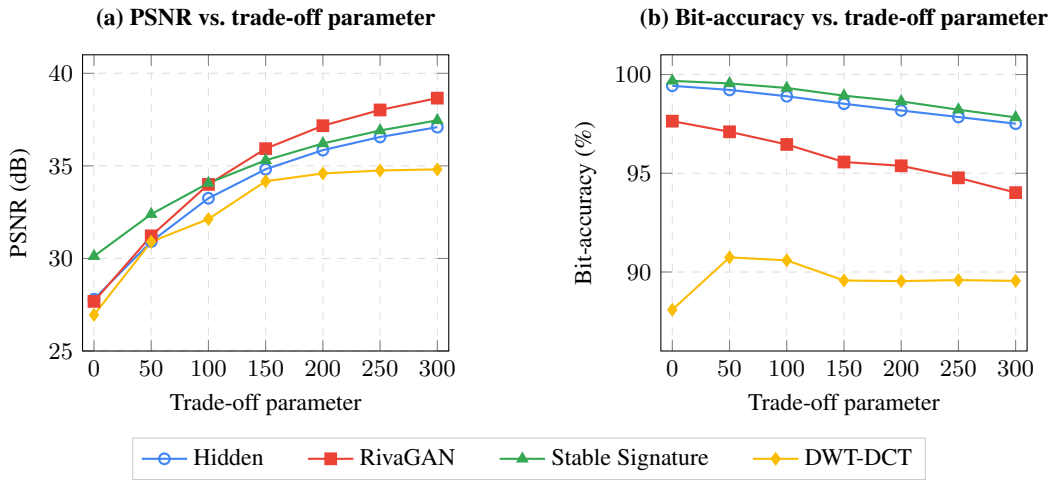


Figure 3: Effect of trade-off parameter λ_d on PSNR and bit accuracy (refinement steps fixed to 100).

B.4 Ablation Study: Impact of Post-Removal Refinement

As described in Section 3.2, the watermark-removed image $\hat{x}_r$ may exhibit reconstruction artifacts from the diffusion regeneration process. We compare two refinement strategies: latent-space perceptual refinement, which optimises in the VAE latent space against a combined pixel and perceptual loss, and score-guided Langevin correction, which iteratively refines the image using a learned score function. Table 8 reports PSNR, SSIM, and MS-SSIM relative to the original watermarked image across four watermarking schemes. Both methods substantially improve visual fidelity over the unrefined attacked image, with latent-space perceptual refinement yielding consistently higher PSNR and SSIM. Importantly, neither method significantly increases the residual bit accuracy on the training data, confirming that the refinement improves perceptual quality without reintroducing the watermark signal.

Table 8: Bit-accuracy, PSNR, SSIM, and MS-SSIM for latent-space perceptual refinement and score-guided Langevin correction compared to watermarked and attacked images.

Scheme	State	Method	Bit-acc	PSNR	SSIM	MS-SSIM
Hidden	Original		57.76	—	—	—
	Watermarked		99.49	37.19	0.9510	0.9918
	Attacked		58.51	23.86	0.6794	0.8635
	Refined	Latent-space perceptual refinement	59.13	30.58	0.8417	0.9620
		Score-guided Langevin correction	59.77	28.21	0.7867	0.9485
RivaGAN	Original		46.41	—	—	—
	Watermarked		98.25	40.56	0.9784	0.9888
	Attacked		57.16	24.21	0.6960	0.8651
	Refined	Latent-space perceptual refinement	67.12	30.42	0.8347	0.9566
		Score-guided Langevin correction	68.45	28.09	0.8005	0.9460
Stable Signature	Original		47.15	—	—	—
	Watermarked		99.42	26.76	0.8570	0.9661
	Attacked		54.38	22.21	0.7276	0.8792
	Refined	Latent-space perceptual refinement	57.96	29.00	0.9090	0.9770
		Score-guided Langevin correction	51.06	28.29	0.9127	0.9764
DWT-DCT	Original		50.25	—	—	—
	Watermarked		86.05	38.00	0.9632	0.9894
	Attacked		49.05	24.65	0.6992	0.8665
	Refined	Latent-space perceptual refinement	49.19	28.79	0.7912	0.9389
		Score-guided Langevin correction	49.10	25.53	0.6973	0.8975

C Additional Experiments: Defense

C.1 Effect of Message Pool Size on Forgery Difficulty

To empirically quantify how message diversity affects forgery performance, we train and evaluate EraseForge on RivaGAN across four datasets with $N = 1{,}000$ samples, progressively increasing the message pool size $K \in \{1, 10, 50, 100\}$. In each setting, the same pool size is used during

both training and evaluation. As shown in Table 9, bit accuracy drops sharply from ∼93–99% at $K = 1$ to ∼68–70% at $K = 100$ across all four evaluation datasets, confirming that a larger message space forces the forgery estimator to generalize across more diverse watermark patterns, substantially degrading its effectiveness. The decline is steepest between $K = 1$ and $K = 10$, with diminishing marginal returns beyond $K = 50$, suggesting that the initial introduction of message diversity is the most disruptive factor for estimator-based attacks. The results are consistent with other estimator-based forgery approaches such as WMCopier (Dong et al., 2025). These results motivate the multi-message defense analyzed in Section 5 and provide the empirical basis for the self-verifiable message mechanism proposed therein.

Table 9: Bit accuracy (%) and PSNR for different number of messages K using RivaGAN. Higher K increases the difficulty of forgery.

	$K = 1$		$K = 10$		$K = 50$		$K = 100$	
Dataset	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc	PSNR	Bit-acc
MS-COCO	31.21	97.13	31.97	82.34	29.81	68.37	28.01	69.77
CelebAHQ	31.09	99.45	34.60	82.23	33.21	67.09	34.86	69.70
ImageNet	26.93	97.71	32.01	81.19	29.70	68.11	32.34	69.56
DiffusionDB	28.15	92.53	32.55	80.44	30.63	68.28	32.82	69.21

D Self-verifiable message: Additional Details

We provide implementation details for the two variants of the self-verifiable message mechanism introduced in Section 5.

D.1 SeedBind: Verifiable Pseudo-random message

For the SeedBind, we instantiate the generation function $g(\cdot)$ using SimHash (Charikar, 2002), a locality-sensitive hashing scheme based on random hyperplane projections. Given a k-bit random seed $\hat{m}$, SimHash computes the remaining $M - k$ bits as follows. The seed bits are first mapped to $\{-1, +1\}$ and then projected through a fixed random weight matrix $\mathbf{W} \in \mathbb{R}^{(M-k)\times k}$ (generated from a shared secret seed):

$$g(\hat{m}) = \mathrm{sign}\big(\mathbf{W} \cdot (2\hat{m} - 1)\big), \tag{10}$$

where the sign function binarises each projection to produce the hash bits. The full message is constructed as $\mathbf{m} = [\hat{m} \; ; \; g(\hat{m})]$.

We evaluate this variant on RivaGAN with a 32-bit message capacity: $k = 16$ bits are sampled uniformly at random and the remaining 16 bits are generated via SimHash, yielding a full 32-bit self-verifiable message. During verification, the detector extracts 32 bits, recovers the first 16 as the seed, recomputes the SimHash projection, and checks bit-level consistency with the last 16 decoded bits. The key advantage of SimHash over hard hashing functions is its tolerance to bit-flip errors: small perturbations in the input seed produce small changes in the output hash, enabling graceful degradation under noisy decoding.

D.1.1 Towards Stronger Verification with Cryptographic Hashing and ECC

While the LSH-based constructions above offer robustness to bit-level noise, they do not provide cryptographic security guarantees. A natural extension is to replace the LSH function $g(\cdot)$ with a cryptographic hash (e.g. SHA-256) and protect the embedded message with Error-Correcting Codes (ECC). Concretely, given an M-bit watermark payload, one can partition it into three segments:

$$\mathbf{m} = [\underbrace{r}_{k \text{ bits}} \; ; \; \underbrace{\mathrm{Hash}(r)}_{l \text{ bits}} \; ; \; \underbrace{\mathrm{ECC}}_{M-k-l \text{ bits}} \;], \tag{11}$$

where r is a k-bit random seed (or content-derived hash), $\mathrm{Hash}(r)$ is a truncated cryptographic hash providing collision resistance and tamper detection, and the remaining $M - k - l$ bits carry an error-correcting code (e.g. BCH or Reed–Solomon) that protects the seed and hash against bit-flip errors introduced by watermark encoding, image compression, or adversarial post-processing. During verification, the ECC is first decoded to recover corrected estimates of r and $\mathrm{Hash}(r)$, after which the hash consistency check is performed. This construction offers stronger forgery resistance than LSH—since cryptographic hashes are non-invertible and exhibit avalanche behavior under even

single-bit changes—at the cost of requiring watermarking schemes that support a sufficiently large payload to accommodate all three segments.

D.2 CONTENTBIND: CONTENT-DEPENDENT MESSAGE

For the ContentBind, we instantiate the feature extractor $\phi(\cdot)$ using DINOv2 (Oquab et al., 2024) (`vit_small_patch14_dinov2.lvd142m`), a self-supervised vision transformer that produces a 384-dimensional feature vector for each input image. The binarization function $h(\cdot)$ is a random hyperplane LSH: a fixed random projection matrix $\mathbf{P} \in \mathbb{R}^{M\times 384}$ (generated from a shared secret seed) maps the feature vector to an M-bit binary code:

$$\mathbf{m} = h\big(\phi(\mathbf{x})\big) = \operatorname{sign}\big(\mathbf{P}\cdot\phi(\mathbf{x})\big). \tag{12}$$

During verification, the verifier recomputes $h(\phi(\mathbf{x}'))$ on the received image $\mathbf{x}'$ and compares it with the decoded watermark message. Because DINOv2 features are semantically rich yet sensitive to image identity, forging a watermark from a source image $\mathbf{x}_{\text{src}}$ onto a visually distinct target $\mathbf{x}_{\text{tgt}}$ causes the recomputed hash to diverge from the decoded message, exposing the forgery.

The projection matrix $\mathbf{P}$ serves as a shared secret between the watermark embedder and verifier. Without knowledge of $\mathbf{P}$, an attacker cannot predict or reverse-engineer the content-dependent message, adding a layer of security beyond the feature extractor itself. We note that the framework is agnostic to the choice of backbone: CLIP and ResNet are equally viable alternatives, trading off between semantic richness and computational cost.

D.3 COMPARATIVE LIMITATIONS OF SEEDBIND AND VARIANT CONTENTBIND

The two variants defend against complementary threat models but carry distinct limitations. Because verification relies solely on the internal consistency between the seed and its hash, Variant 1 is effective against estimator-based attacks that must generalize across diverse messages during training. However, it is inherently vulnerable to residual-based attacks (e.g. WMForger (Souček et al., 2025b)), which copy the watermark signal—including a structurally valid seed–hash pair—from a single observation. Additionally, SimHash is a linear projection: given a sufficient number of observed (seed, hash) pairs, an adversary could in principle recover the projection matrix $\mathbf{W}$, undermining the shared secret. On the other hand, Variant 1 is robust to arbitrary image transformations, since verification depends only on the decoded message bits and not on the visual content of the image. By binding the message to the image's visual content, Variant 2 neutralizes both estimator-based and residual-based attacks, as demonstrated in Table 5. However, this content-dependence introduces a sensitivity to image edits that significantly alter the semantic features extracted by the backbone (e.g. heavy cropping, style transfer, or large inpainting regions). Such edits may cause the recomputed hash $h(\phi(\mathbf{x}'))$ to diverge from the decoded message even for legitimately watermarked images, leading to false rejections. Moreover, the watermarking process itself perturbs the image, requiring that the chosen feature extractor produces sufficiently stable features under watermark-level distortions—an empirical property rather than a theoretical guarantee.

Summary. SeedBind is the more practical choice when the watermarked image may undergo significant post-processing but the threat model is limited to estimator-based attacks when the underlying watermarking approach has a higher degree of content-agnosticity. As noted in prior literature (Souček et al., 2025b; Yang et al., 2024a), watermarking approaches that are more content-dependent are not vulnerable to residual-based forgery attacks. ContentBind provides stronger protection against a broader class of attacks even when the underlying watermarking approach is vulnerable at the cost of reduced tolerance to semantic-level image modifications.

E LIMITATIONS

The LSH-based constructions (SimHash for SeedBind, binarization function for ContentBind) proposed in this work demonstrate the feasibility of self-verifiable messages but are not cryptographically secure. Replacing them with stronger primitives—such as cryptographic hash functions with error-correcting codes, as outlined in Appendix D.1.1—would provide formal guarantees. Designing watermarking schemes that jointly optimize for payload capacity, perceptual quality, and cryptographic verifiability is a promising direction for building provenance systems that are resilient to both removal and forgery attacks. Our experiments focus on four watermarking schemes (DWT-DCT, HiDDeN, RivaGAN, and Stable Signature) that support message capacities of 32–48 bits. Moreover,

emerging watermarking systems with substantially larger payloads (e.g. 256 bits or more) may be more amenable to the cryptographic hash and ECC-based defenses discussed in Section 5 and Appendix D.1.1, as they offer sufficient bit budget to accommodate seed, hash, and redundancy segments without sacrificing detection reliability. Evaluating EraseForge and the proposed defenses on such higher-capacity schemes is an important direction for future work.

While forged watermarks produced by EraseForge achieve high bit accuracy under clean conditions, they may exhibit reduced robustness compared to genuinely embedded watermarks when subjected to post-processing distortions such as JPEG compression, Gaussian blur, or noise injection (see Table 2). This robustness gap could serve as a statistical signal for defenders to distinguish forged from genuine watermarks, though exploiting it reliably at scale remains an open problem.

Although the approach discussed in this paper relies on the success of watermark removal, current watermarking schemes are vulnerable to some form of removal attack (Liu et al., 2025). Recent theoretical work has established that diffusion-based image transformations cause the mutual information between the watermark payload and the processed image to vanish along the diffusion trajectory (Ni et al., 2025; Guo et al., 2026; Qi et al., 2026). Specifically, because watermark signals reside off the natural image manifold by construction, diffusion-based regeneration progressively contracts these off-manifold perturbations, making watermark recovery information-theoretically impossible—not merely practically difficult—under sufficient transformation. These results suggest that the watermark removal assumption underlying EraseForge is well-founded across a broad class of existing watermarking schemes, as no current embedding-based approach can provably resist diffusion-based removal.

F Broader Impact

Watermarking is increasingly relied upon as a trust infrastructure for AI-generated content, underpinning regulatory compliance (e.g. the EU AI Act, U.S. Executive Orders, California SB 942) and platform-level provenance systems. By exposing that current watermarking defenses—including the multi-message strategy—can be circumvented with modest data requirements, our work provides an early warning to the research community and industry practitioners before these vulnerabilities are exploited in the wild. The self-verifiable message defenses we propose (SeedBind and ContentBind) offer concrete, architecture-agnostic countermeasures that can be integrated into existing watermarking pipelines without modification of the underlying encoder or decoder. More broadly, this work contributes to the adversarial evaluation methodology that is essential for building trustworthy provenance systems: understanding how forgery attacks succeed is a prerequisite for designing systems that are resilient to them.

The tension between watermark robustness and forgery resistance reflects a fundamental challenge in the design of media provenance systems. Our findings suggest that message-level defenses alone may be insufficient, and that cryptographically grounded verification—where the watermark message is structurally tied to the image content or an internal consistency constraint—offers a more principled path forward. We hope this work encourages the community to adopt red-teaming as a standard component of watermarking system evaluation and to pursue tighter integration between watermarking and cryptographic verification in future designs.

G QUALITATIVE EXAMPLES

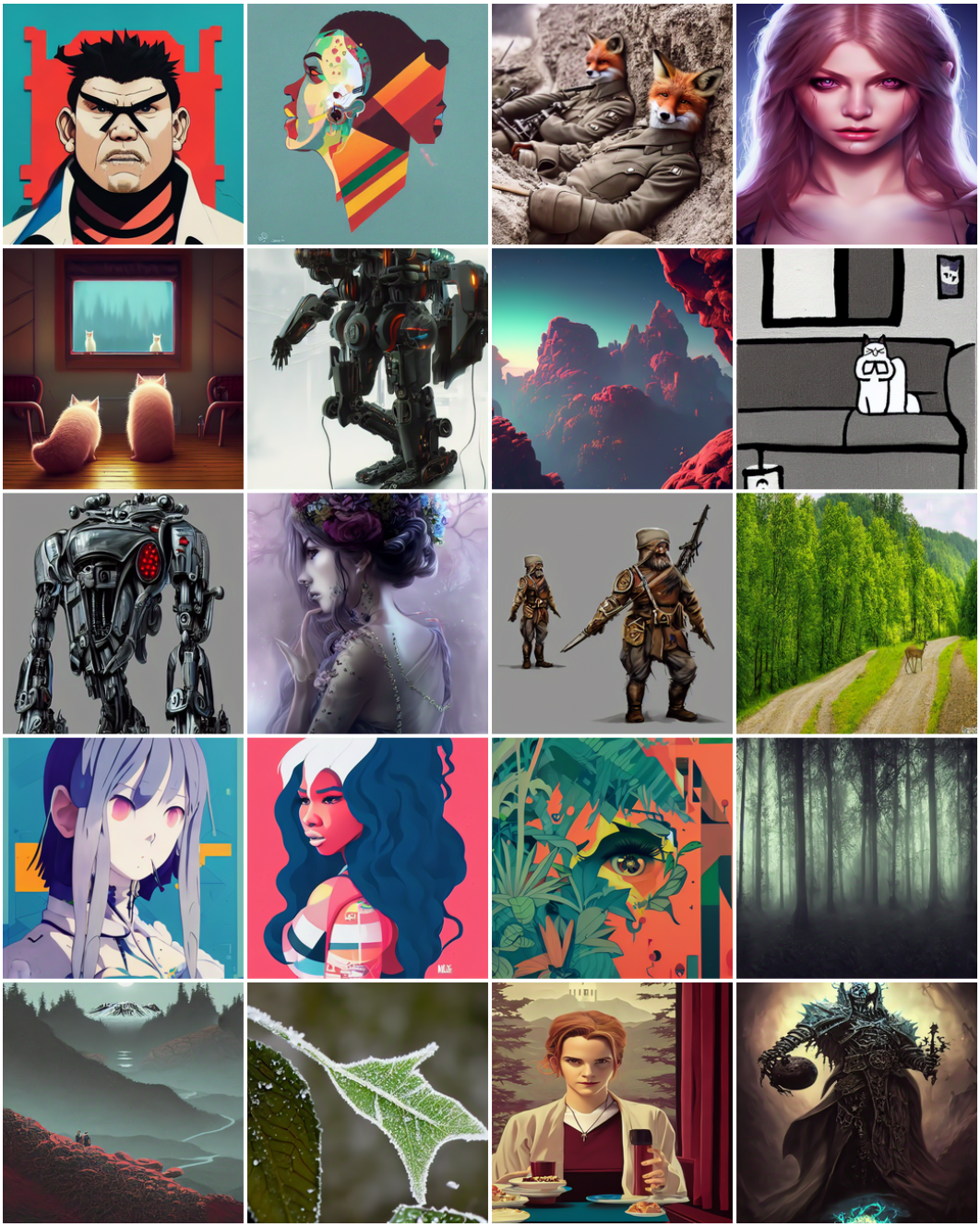


Figure 4: Qualitative examples of forged watermarked images produced by EraseForge using RivaGAN on the DiffusionDB dataset.

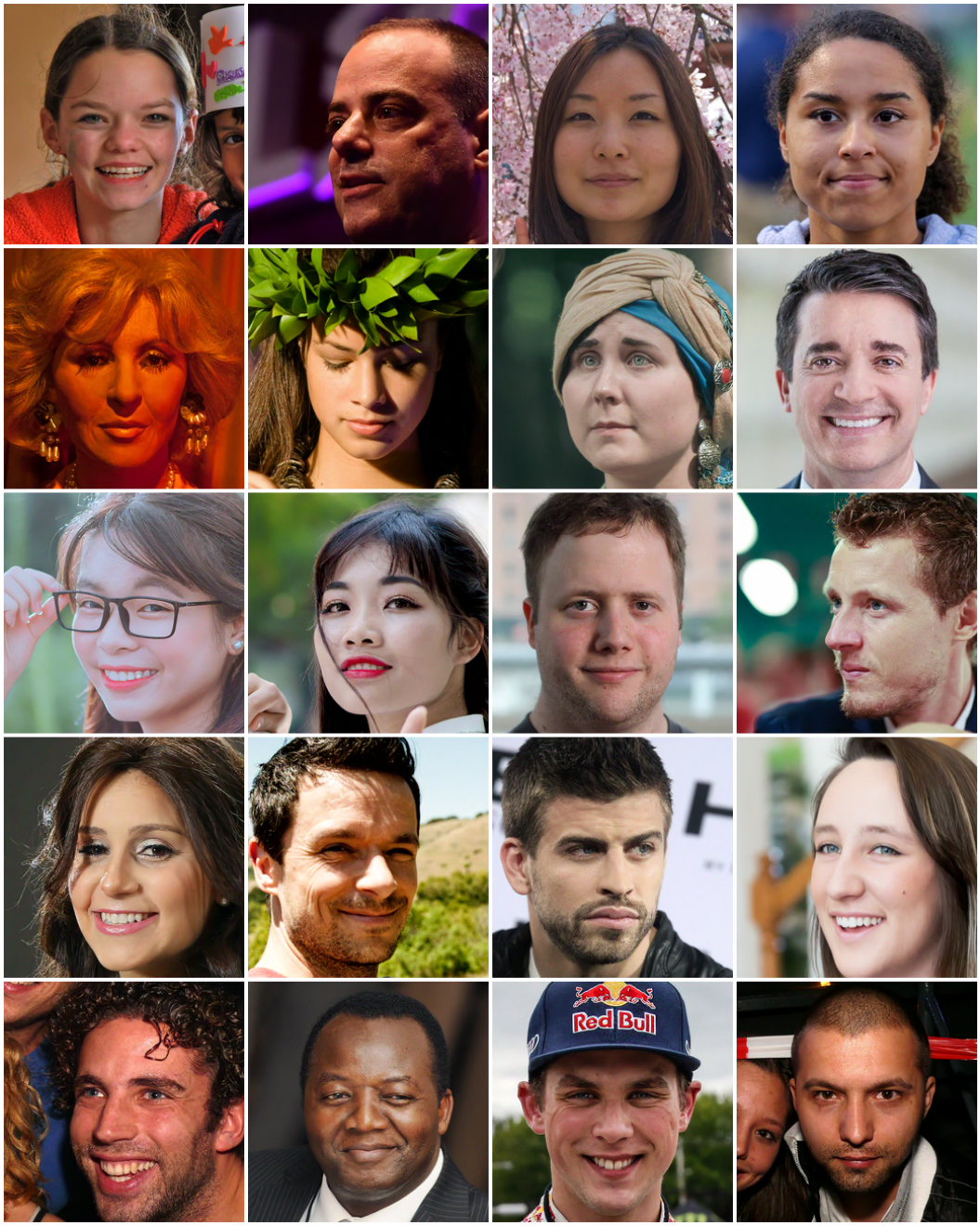


Figure 5: Qualitative examples of forged watermarked images produced by EraseForge using RivaGAN on the CelebAHQ dataset.

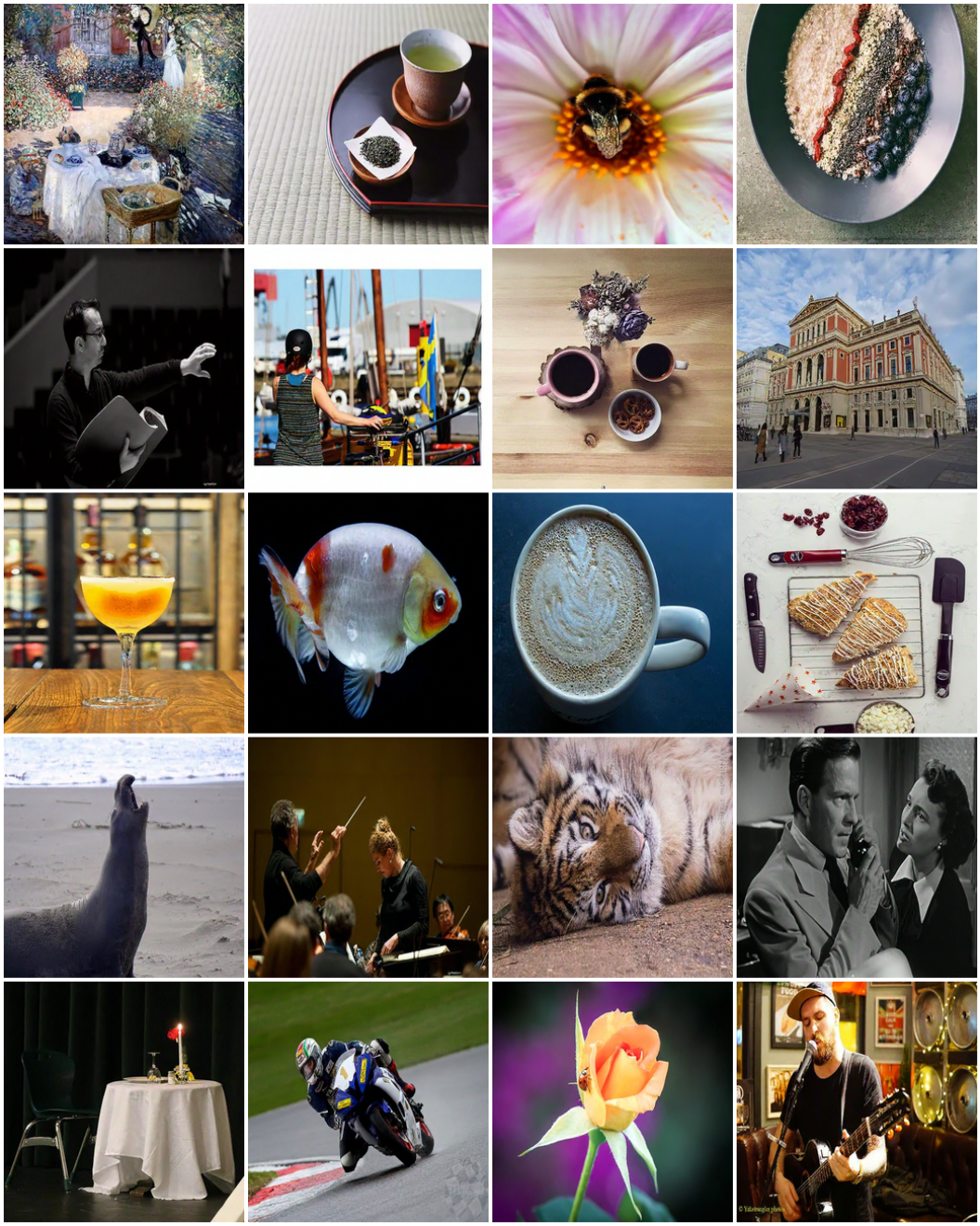


Figure 6: Qualitative examples of forged watermarked images produced by EraseForge using RivaGAN on the ImageNet dataset.

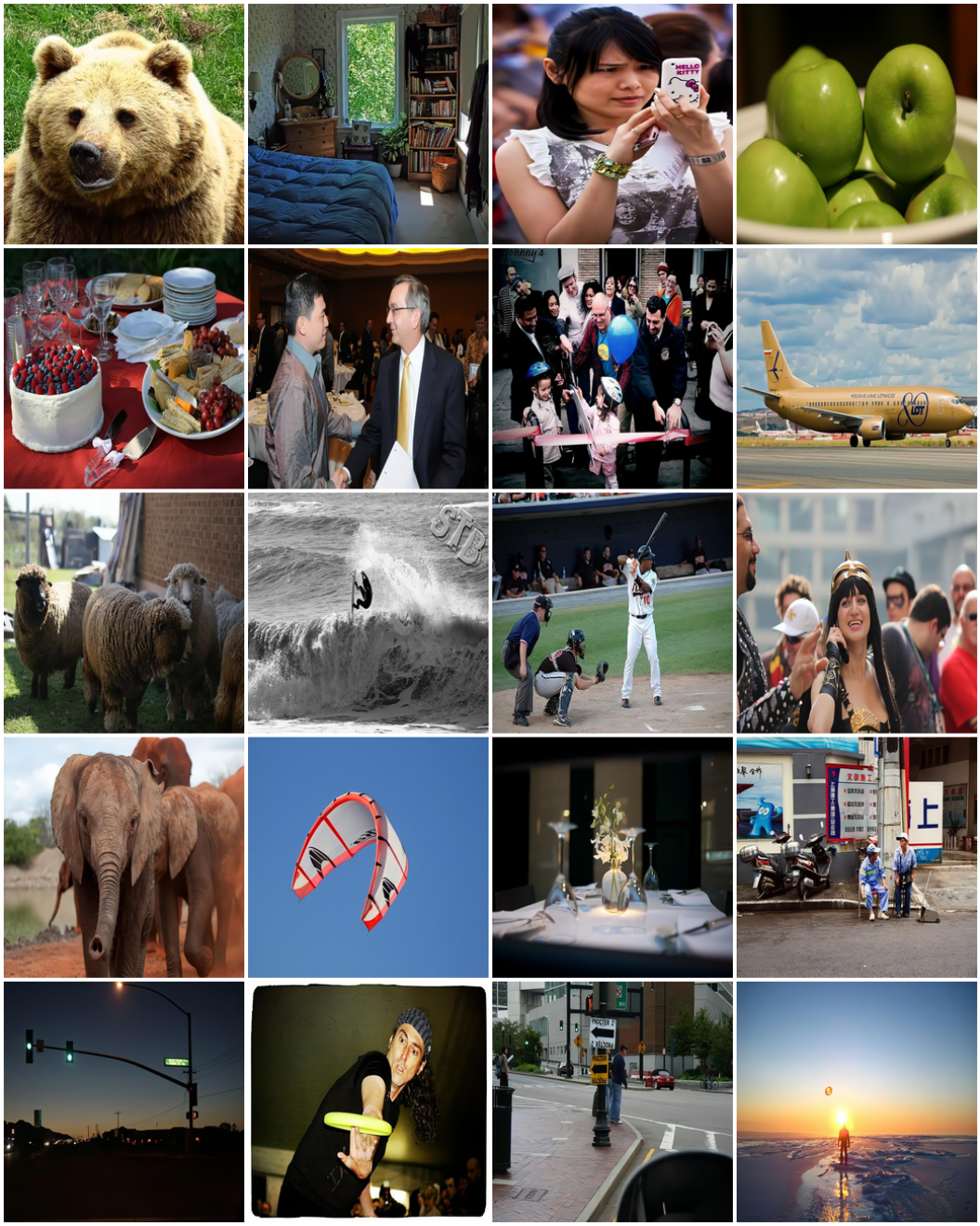


Figure 7: Qualitative examples of forged watermarked images produced by EraseForge using RivaGAN on the MSCOCO dataset.