SWIFTREGEN: EFFICIENT IMAGE WATERMARK EVASION BEYOND OUTPUT IMITATION AND EVIDENCE REMOVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Robust invisible image watermarking relies on embedded evidence that remains detectable under post-processing and adversarial manipulation. Regeneration attacks can suppress this evidence, but iterative denoising incurs substantial computational cost. We present **SwiftRegen**, an efficient watermark evasion attack that replaces iterative regeneration with a single-pass operator learned offline from clean images and their regenerated counterparts, without watermark supervision, detector queries, or keys. Our study reveals two key insights. First, faithfully reproducing the regeneration output does not guarantee effective removal transfer: a full-image predictor achieves higher output fidelity yet weaker removal than a bounded residual predictor, so learning the regeneration-induced update transfers removal more effectively. Second, watermark evidence can survive while detection fails. For schemes that depend on correspondence between surviving evidence and the detector-side reference, disrupting this correspondence can defeat detection without removing the evidence. This motivates **Watermark Carrier Desynchronization (WCD)**, which uses mild cropping and resizing to disrupt synchronization while largely preserving visual content. SwiftRegen combines the learned operator with WCD, and across eleven watermarking schemes achieves a mean TPR@1% FPR of 0.080 and a $72\times$ speedup over CtrlRegen. Our results show that robust watermark detection depends on both evidence persistence and synchronization.

1 INTRODUCTION

Invisible image watermarking has become an important mechanism for content attribution, provenance verification, and copyright protection of AI-generated media (Cox et al., 2008; An et al., 2024). A robust watermark must remain detectable not only after benign post-processing, but also under deliberate attempts to invalidate it, so existing removal attacks largely focus on weakening, suppressing, or reconstructing away watermark evidence. Recent regeneration-based attacks demonstrate that powerful generative priors can remove a broad range of watermarks without access to detector outputs or watermark keys (Zhao et al., 2024; Liu et al., 2025). Their effectiveness, however, comes at a substantial computational cost: iterative regeneration must execute a full denoising trajectory for every attacked image, for example 50 network evaluations and approximately 3.27 s per image for CtrlRegen on an RTX 2080 Ti (Figure 1(a)).

This raises a practical question: must the expensive regeneration process be repeated online, or can its watermark removal behavior be learned once and reused? A natural approach is to train a feed-forward model to imitate the regeneration output as faithfully as possible. Surprisingly, this does not reliably transfer the behavior we actually care about: a full-image predictor can achieve higher fidelity to the regeneration output while inheriting substantially less of its watermark removal capability (Section 6.1). In contrast, learning the bounded residual update induced by regeneration transfers removal much more effectively, suggesting that effective amortization should capture what regeneration changes rather than merely reproduce what it outputs.

This enables us to replace costly iterative regeneration with a single-pass operator learned offline from clean images and their regenerated counterparts, requiring no target-watermark supervision, detector queries, watermark keys, or scheme-specific adaptation at deployment. It reframes regeneration-based

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.

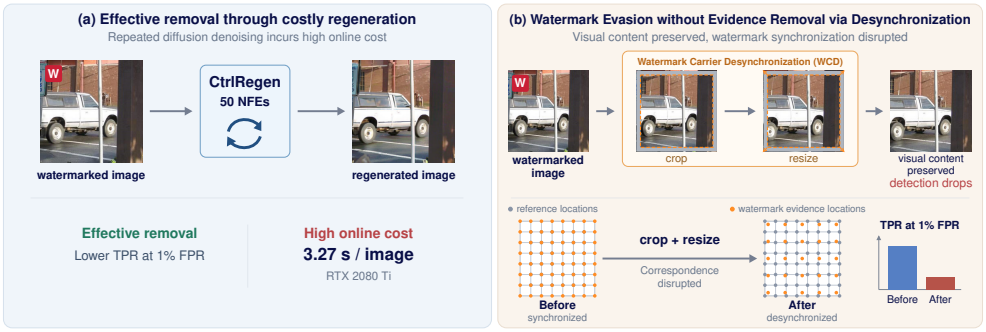


Figure 1: **Motivation: regeneration cost and carrier desynchronization.** (a) CtrlRegen suppresses watermark detection but requires 50 NFEs and 3.27 s per image on an RTX 2080 Ti. (b) Mild crop-and-resize preserves visible content while re-indexing surviving carrier evidence relative to a fixed detector reference. The lower diagrams illustrate this correspondence change.

removal as a transferable transformation rather than an iterative procedure executed from scratch for every image.

Our second observation challenges a more fundamental assumption: successful evasion does not necessarily require eliminating the embedded evidence. For synchronization-dependent schemes, watermark evidence can survive a transformation while detection fails because its correspondence with the detector-side reference has been disrupted, so evidence persistence alone is insufficient when detection also depends on how that evidence is aligned with a reference. This exposes synchronization as a distinct attack surface. We therefore introduce **Watermark Carrier Desynchronization (WCD)**, which uses mild geometric re-indexing to disrupt this correspondence while largely preserving visual content (Figure 1(b)). Restoring the known crop geometry recovers detection in our WCD experiments, providing direct evidence that watermark evidence can survive while detection fails (Appendix E).

Building on these observations, SwiftRegen combines the learned operator with WCD (Figure 2). We evaluate the complete pipeline and its components across eleven watermarking schemes. Our contributions are:

- **Learning Removal beyond Output Imitation.** We show that the removal behavior of iterative regeneration can be learned from clean images and their regenerated counterparts and transferred to a single-pass operator without watermark supervision, detector queries, or keys. Higher fidelity to the regeneration output does not guarantee stronger removal transfer; learning its bounded residual update transfers removal more effectively.
- **Evidence Survives, Detection Fails.** We show that watermark evasion does not necessarily require eliminating the embedded evidence. For schemes that depend on synchronization, disrupting correspondence with the detector's reference can defeat detection while evidence remains. This motivates WCD and identifies synchronization as a distinct attack surface. The results indicate that robust watermarking must protect both evidence persistence and synchronization.
- **Effective and Efficient Watermark Evasion.** Across eleven watermarking schemes, SwiftRegen achieves a mean TPR@1% FPR of 0.080 and a 72× speedup over CtrlRegen. Matched ablations demonstrate the complementary effects of the learned operator and WCD, while additional experiments assess image quality and the recoverability of watermark evidence.

2 Related Work

Image watermarking. Invisible watermarking methods include classical and learning-based post-hoc watermarking methods such as DwtDCT, HiDDeN, RivaGAN, StegaStamp, TrustMark, Robust-Wide, and VINE (Al-Haj, 2007; Zhu et al., 2018; Zhang et al., 2019; Tancik et al., 2020; Bui et al.,

2025; Hu et al., 2024; Lu et al., 2025), as well as methods that embed watermarks during generation, such as Stable Signature, Tree-Ring, Gaussian Shading, and PRC (Fernandez et al., 2023; Wen et al., 2023; Yang et al., 2024; Gunn et al., 2025). Benchmarking work emphasizes that robustness must be measured against both conventional perturbations and stronger removal attacks (An et al., 2024).

Watermark removal. Detector-agnostic attacks include compression, filtering, geometric transformations, and reconstruction using image priors. VAE and diffusion attacks modify an image through reconstruction or regeneration (Zhao et al., 2024). DIP fits an image prior separately for each input (Ulyanov et al., 2018; Liang et al., 2025). CtrlRegen uses controllable regeneration (Liu et al., 2025), while RAVEN uses novel-view synthesis (Shamshad et al., 2026). UnMarker optimizes objectives that do not require access to the target detector (Kassis & Hengartner, 2025). Related work also studies optimization for adaptive watermark attacks (Lukas et al., 2024).

Learning efficient removal. MarkNull targets the dependence between latent representations and embedded initial noise, and MarkNull-A provides an amortized variant that requires a single forward pass (Cao et al., 2026). Related purification methods also learn efficient approximations to iterative processing (Lei et al., 2025). SwiftRegen learns from clean images and their CtrlRegen outputs. We study whether fidelity to those outputs predicts watermark removal performance and show that bounded residual prediction transfers removal more effectively than full-image prediction.

Geometry and synchronization. Geometric attacks and synchronization are established concerns in watermarking (Cox et al., 2008; An et al., 2024). Cropping and resizing change the correspondence between watermark evidence and the detector's reference. WCD studies this dependence alongside learned watermark removal. Our registration experiments directly test whether restoring that correspondence recovers detection.

3 Threat Model

We consider an image provider that embeds invisible watermarks and an adversary seeking to evade detection while preserving image content. The adversary has no access to the target detector, watermark keys, or internal details of the watermarking system. Neither training nor inference uses detector gradients, scores, decoded messages, or binary responses; detectors are used only for evaluation. Offline training uses publicly available unwatermarked images and their outputs from a fixed CtrlRegen model. At inference, the attack receives only the watermarked image x_w and uses the same trained operator and fixed WCD settings across all evaluated schemes. It does not require the corresponding unwatermarked image. Paired clean references are used only where explicitly specified in diagnostic experiments.

4 SwiftRegen

Given a watermarked image x_w, SwiftRegen first applies a learned residual operator $\mathcal{A}_\theta$, then performs chromatic restoration $\mathcal{C}$, and finally applies WCD $\mathcal{W}$ (Figure 2). The complete pipeline is

$$x_0 = \mathcal{A}_\theta(x_w), \qquad x_c = \mathcal{C}(x_0; x_w), \qquad \hat{x} = \mathcal{W}(x_c). \tag{1}$$

The learned operator transfers removal behavior from iterative regeneration, chromatic restoration corrects brightness and color shifts, and WCD disrupts synchronization.

4.1 Learning Bounded Residual Updates

Let $\mathcal{R}$ denote the fixed CtrlRegen process. Applying $\mathcal{R}$ directly to each watermarked input requires iterative diffusion sampling. We instead train $\mathcal{A}_\theta$ offline using clean images x and their regenerated counterparts $\mathcal{R}(x)$ (Figure 2(a)). The learned operator is then applied to watermarked inputs without detector feedback or watermark supervision.

Higher fidelity to $\mathcal{R}(x)$ does not guarantee stronger watermark removal (Section 6.1). We therefore retain the input through an identity connection and predict the change induced by regeneration. The target update is

$$r^*(x) = \mathcal{R}(x) - x. \tag{2}$$

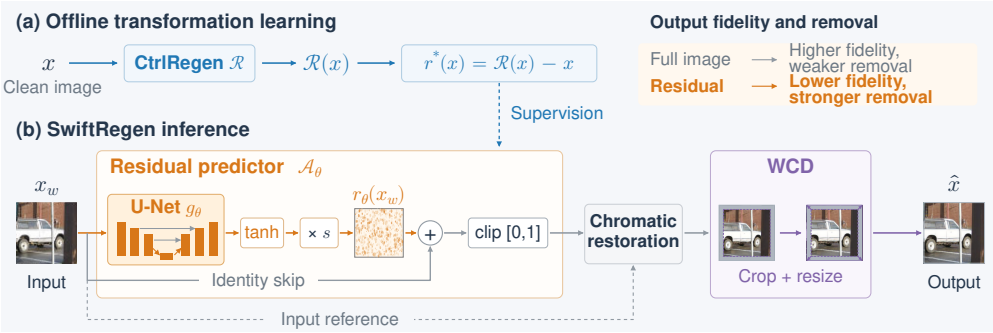


Figure 2: **SwiftRegen workflow.** (a) Clean images and their regenerated counterparts provide supervision for learning residual updates offline. (b) The learned residual operator is followed by chromatic restoration and WCD. The inset summarizes the comparison in Section 6.1: full-image prediction gives higher fidelity to the regeneration output but weaker removal than bounded residual prediction. The residual texture is schematic.

The network g_θ predicts a bounded residual, which is added to the input and clipped to the valid image range:

$$\begin{aligned} r_\theta(x) &= s \tanh g_\theta(x), \\ \mathcal{A}_\theta(x) &= \mathrm{clip}(x + r_\theta(x), 0, 1)\,. \end{aligned} \tag{3}$$

The identity connection carries the input content, while s controls the maximum residual magnitude. We use $s = 0.5$ during pretraining and $s = 0.8$ during refinement. The predictor g_θ is a residual U-Net with four scales and channel widths of 48, 96, 192, and 384. Architectural and training details are provided in Appendix B.

The training objective contains three groups. The operator loss supervises the predicted residual and output. The structure loss encourages perceptual, edge, and semantic similarity. The distribution loss constrains residual energy and image and residual statistics. The component losses and their weights are given in Appendix B.2. Inference requires one network evaluation and no iterative diffusion sampling.

Chromatic restoration. The learned operator can introduce brightness and color shifts. Before WCD, $\mathcal{C}$ uses the watermarked input as a reference to adjust local color statistics, correct local mean differences, and apply mild unsharp masking. The formula and parameters are given in Appendix B.

4.2 Watermark Carrier Desynchronization

Some watermark detectors require the embedded evidence to remain aligned with a reference. Redundancy can help evidence withstand local corruption, but it does not by itself preserve this alignment after geometric changes. WCD disrupts the correspondence through cropping and resizing (Figure 1(b)).

Spatial correspondence. To illustrate this dependence, let $z(u)$ denote watermark evidence at normalized spatial coordinates $u \in [0,1]^2$, and let $k(u)$ denote a detector reference. A simple correspondence statistic is

$$\langle z, k \rangle = \int_{[0,1]^2} z(u)k(u)\,du. \tag{4}$$

A crop with retained side fraction q and origin t maps a retained location u to output coordinates p. After resizing, the evidence is approximately

$$u \longmapsto p = \frac{u - t}{q}, \qquad z_{t,q}(p) \approx z(t + qp), \qquad t \sim \mathcal{U}([0, 1 - q]^2). \tag{5}$$

This expression isolates the coordinate change from discrete resampling. If the detector uses its original reference without compensating for the transformation, it evaluates

$$\langle z_{t,q}, k \rangle \approx \int_{[0,1]^2} z(t + qp)k(p)\, dp. \tag{6}$$

The evidence and reference are now compared at different corresponding locations. The crop origin changes position, and resizing changes scale and sampling. Appendix B.5 describes the resulting image spectrum. The additional experiments in Appendix E measure the effects on VAE latents and noise recovered by inversion.

Implementation. WCD samples a crop origin uniformly and uses bilinear interpolation to restore the original resolution:

$$\mathcal{W}_{t,q}(y)(p) = \mathrm{Bilinear}(y, t + qp)\,, \qquad t \sim \mathcal{U}([0, 1 - q]^2), \tag{7}$$

We use $q = 0.90$ by default, retaining 90% of each spatial dimension, or 81% of the image area before resizing. Section 5.4 compares WCD alone, the learned operator with chromatic restoration, and the complete pipeline using matched crop seeds.

5 Experiments

5.1 Experimental Setup

The main benchmark uses 1,000 image-caption pairs sampled from COCO2017 validation (Lin et al., 2014). For schemes that embed watermarks during generation, we use the sampled COCO captions as prompts to generate the watermarked evaluation images. UnMarker and full MarkNull are evaluated on 100 images per watermark. We evaluate eleven watermarking schemes and use the same trained SwiftRegen operator and fixed attack settings across schemes unless stated otherwise.

We evaluate eleven watermarking methods: DwtDct (Al-Haj, 2007), DwtDctSvd (Navas et al., 2008), HiDDeN (Zhu et al., 2018), RivaGAN (Zhang et al., 2019), StegaStamp (Tancik et al., 2020), Stable Signature (Fernandez et al., 2023), VINE (Lu et al., 2025), RobustWide (Hu et al., 2024), Gaussian Shading (Yang et al., 2024), Tree-Ring (Wen et al., 2023), and PRC (Gunn et al., 2025). Reconstruction and regeneration baselines include VAE-B (Ballé et al., 2018), VAE-C (Cheng et al., 2020), and Regen (Zhao et al., 2024). We also compare against CtrlRegen (Liu et al., 2025), DIP (Ulyanov et al., 2018; Liang et al., 2025), UnMarker (Kassis & Hengartner, 2025), and the MarkNull attacks (Cao et al., 2026).

Detector thresholds are calibrated on held-out clean references and fixed across attacks. We report TPR@1% FPR to measure detection after attack; lower values indicate more effective evasion. LPIPS (Zhang et al., 2018), DINOv2 (Oquab et al., 2024), CLIP (Radford et al., 2021), and FID (Heusel et al., 2017) assess perceptual differences, semantic similarity, and distributional differences. Aligned PSNR and SSIM (Wang et al., 2004) are reported in Appendix F. Calibration, sample selection, and implementation details are provided in Appendices A and B.

5.2 Main Removal Evaluation

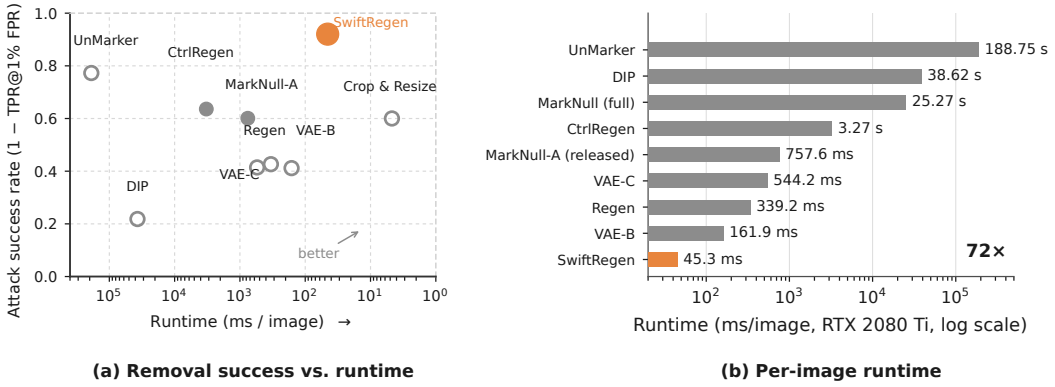
Table 1 compares watermark removal attacks under fixed detector thresholds, including BM3D denoising (Dabov et al., 2007) and the recent MarkNull attack. SwiftRegen achieves the lowest mean TPR across the eleven schemes, with 0.080 compared with 0.367 for CtrlRegen, 0.400 for Crop & Resize, 0.399 for MarkNull-A, and 0.233 for full MarkNull. The relative performance varies by scheme. Crop & Resize strongly reduces detection for VINE, RobustWide, Gaussian Shading, and PRC, but leaves RivaGAN and Stable Signature almost fully detectable and Tree-Ring at 0.956. SwiftRegen reduces their TPRs to 0.224, 0.144, and 0.149, respectively. CtrlRegen remains stronger on individual schemes, including RivaGAN and Stable Signature. Matched component comparisons in Section 5.4 examine the contributions of the learned operator and WCD.

5.3 Removal Efficiency

On an RTX 2080 Ti at batch size 1, the complete SwiftRegen pipeline takes 45.3 ms per image, including chromatic restoration and WCD. CtrlRegen takes 3.27 s, corresponding to a $72\times$ speedup

Table 1: **Watermark removal across eleven schemes.** Post-attack TPR@1% FPR; lower is better, and the best values are bolded. SwiftRegen denotes the complete pipeline. Each scheme uses 1,000 images, except UnMarker and full MarkNull, which use 100 each. † The MarkNull-A result for RobustWide uses a separate image set. Protocol details are provided in Appendix A.

Attack	*DwtDct*	*DwtDctSvd*	*HiDDeN*	*RivaGAN*	*StegaStamp*	*Stable Sig.*	*VINE*	*RobustWide*	*GS*	*Tree-Ring*	*PRC*	*Mean*
Brightness	0.209	0.514	0.360	0.992	0.996	1.000	1.000	0.996	1.000	0.977	1.000	0.822
Contrast	0.204	0.425	0.179	0.997	0.998	1.000	1.000	0.998	1.000	0.991	1.000	0.799
JPEG	0.018	0.699	0.005	0.984	1.000	0.996	1.000	1.000	1.000	0.930	0.946	0.780
Blur	0.148	0.998	0.010	1.000	1.000	0.981	1.000	1.000	1.000	0.874	0.999	0.819
Noise	0.610	0.995	0.005	1.000	1.000	0.999	1.000	1.000	1.000	0.981	0.905	0.863
BM3D	0.185	0.996	0.007	0.998	1.000	0.994	1.000	1.000	1.000	0.538	0.979	0.791
Center Crop	0.878	0.998	0.028	0.999	1.000	1.000	**0.000**	1.000	1.000	0.999	1.000	0.809
Random Crop	0.777	0.935	0.026	0.998	1.000	1.000	0.119	1.000	1.000	0.999	1.000	0.805
Crop & Resize	0.090	0.286	0.108	0.999	0.829	1.000	0.004	0.001	0.102	0.956	0.023	0.400
VAE-B	0.007	0.407	0.005	0.057	1.000	0.791	1.000	0.961	1.000	0.450	0.790	0.588
VAE-C	0.003	0.302	0.005	0.040	1.000	0.835	1.000	0.998	1.000	0.430	0.830	0.586
Regen	0.005	0.296	0.005	0.173	0.999	0.028	1.000	1.000	1.000	0.920	0.950	0.580
DIP	0.148	0.654	0.014	0.815	1.000	0.993	1.000	1.000	1.000	0.979	0.999	0.782
CtrlRegen	0.013	**0.030**	0.005	**0.004**	0.688	0.022	0.986	0.811	1.000	0.305	0.174	0.367
UnMarker (w/ Crop)	**0.000**	0.090	**0.000**	0.970	0.550	0.820	0.020	**0.000**	**0.040**	**0.000**	0.010	0.227
MarkNull-A	0.007	0.678	0.001	0.136	1.000	0.011	0.664	0.998†	0.504	0.380	0.014	0.399
MarkNull (full)	**0.000**	0.060	**0.000**	0.040	0.900	**0.010**	0.940	0.180	0.170	0.240	0.020	0.233
SwiftRegen	0.008	0.063	0.005	0.224	**0.206**	0.144	0.007	0.004	0.060	0.149	**0.006**	**0.080**



(a) Removal success vs. runtime **(b) Per-image runtime**

Figure 3: **Watermark evasion and runtime.** (a) Attack success rate, defined as $1-\text{TPR@1\% FPR}$, versus per-image runtime; higher and further right is better. SwiftRegen, CtrlRegen, and MarkNull-A share the controlled RTX 2080 Ti protocol; other points use archived timings. (b) Compute-only runtime on RTX 2080 Ti at batch size 1. SwiftRegen takes 45.3 ms per image compared with 3.27 s for CtrlRegen, a 72× speedup.

(Figure 3). Full MarkNull uses iterative optimization and takes 25.3 s per image on the same hardware, more than two orders of magnitude slower than SwiftRegen. The released MarkNull-A pipeline takes 758 ms. Appendix B.6 specifies the timing protocol, and Appendix C examines how residual strength affects removal and image quality.

5.4 Effects of the Learned Operator and WCD

Figure 4 compares WCD only, SwiftRegen without WCD ($\mathcal{A}_\theta + \mathcal{C}$), and the complete pipeline on an independent subset of 1,000 images per scheme, sharing the same crop seed for each image across the

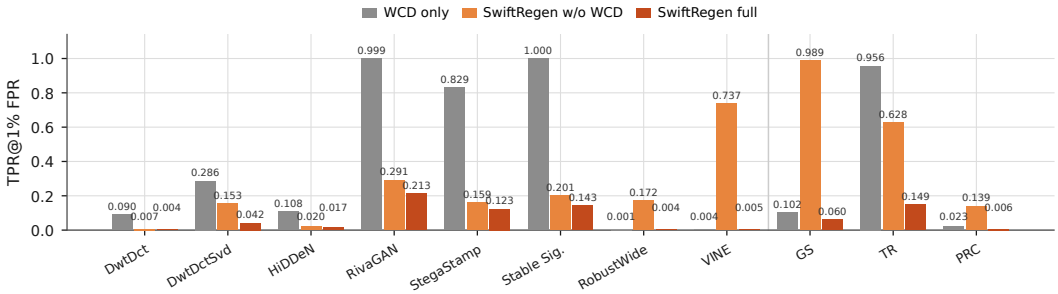
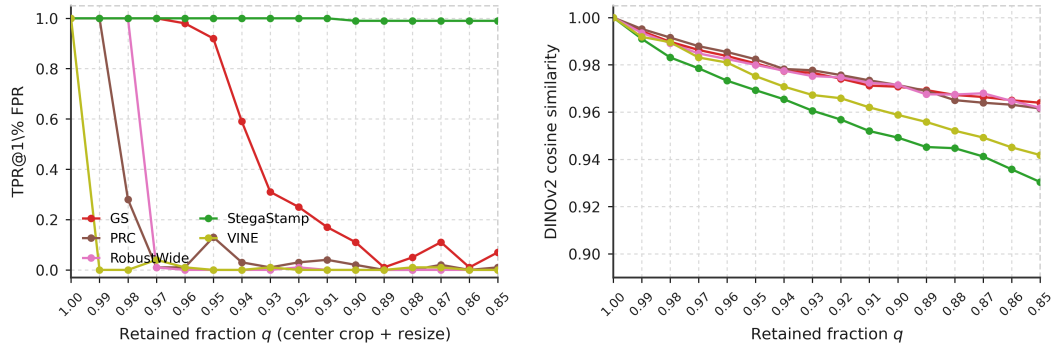


Figure 4: **Effects of the learned operator and WCD.** TPR@1% FPR for WCD only, SwiftRegen without WCD ($\mathcal{A}_\theta + \mathcal{C}$), and the complete pipeline, evaluated on an independent subset of 1,000 images per scheme. The WCD configurations use matched crop seeds.



(a) Detection after WCD **(b) Semantic preservation**

Figure 5: **WCD strength and semantic similarity.** Centered cropping and resizing with decreasing retained side fraction q (stronger intervention toward the right): (a) TPR@1% FPR; (b) DINOv2 cosine similarity to the original watermarked image. Each scheme uses 100 images. The sweep uses centered crops; the default pipeline uses random crop origins.

WCD branches. WCD alone strongly reduces detection for Gaussian Shading, VINE, RobustWide, and PRC. Adding the learned operator with chromatic restoration improves removal for RivaGAN, StegaStamp, and Stable Signature. For Tree-Ring, the combination reduces detection more effectively than either configuration alone. These results show complementary effects across schemes. The configuration without WCD includes chromatic restoration, whose effect on detection is reported separately in Table 7.

WCD strength and image semantics. Figure 5 evaluates centered cropping and resizing on 100 watermarked images per scheme. As the retained side fraction decreases, detection falls sharply for PRC, VINE, and RobustWide and more gradually for Gaussian Shading. StegaStamp remains highly detectable. At $q = 0.90$, DINOv2 cosine similarities remain approximately 0.95–0.97 while detection is low for four of the five schemes. The sweep shows that WCD can reduce detection while retaining high semantic similarity. For StegaStamp, the benefit of adding the learned operator is established by the component comparison in Figure 4. Additional experiments with different training seeds and crop settings are reported in Appendix C.

5.5 Image Fidelity and Qualitative Results

Table 2 reports image quality after attack. Chromatic restoration reduces LPIPS from 0.058 to 0.045 before WCD. The complete pipeline has LPIPS 0.188 relative to the unaligned watermarked input. DINOv2 and CLIP cosine similarities exceed 0.90 for all reported methods, indicating high semantic

Table 2: **Image quality after attack.** Metrics use the original watermarked input as the reference, except UnMarker LPIPS, which uses a geometrically aligned reference. DINOv2 and CLIP report mean cosine similarity; FID compares feature distributions. Sample counts follow Table 1, and metric settings are given in Appendix A.2.

Attack	LPIPS ↓	DINOv2 ↑	CLIP ↑	FID ↓
VAE-B	0.057	0.956	0.923	8.86
VAE-C	0.052	0.948	0.921	8.25
Regen	0.087	0.906	0.925	6.25
DIP	0.014	0.988	0.959	1.29
CtrlRegen	0.128	0.931	0.922	7.45
UnMarker (w/ Crop)	0.087	0.921	0.915	45.02
MarkNull-A	0.066	0.954	0.952	8.91
MarkNull (full)	0.094	0.914	0.936	26.21
SwiftRegen ($\mathcal{A}_\theta$ only)	0.058	0.918	0.917	18.75
SwiftRegen ($\mathcal{A}_\theta + \mathcal{C}$)	0.045	0.932	0.937	15.39
SwiftRegen full	0.188	0.912	0.920	16.52

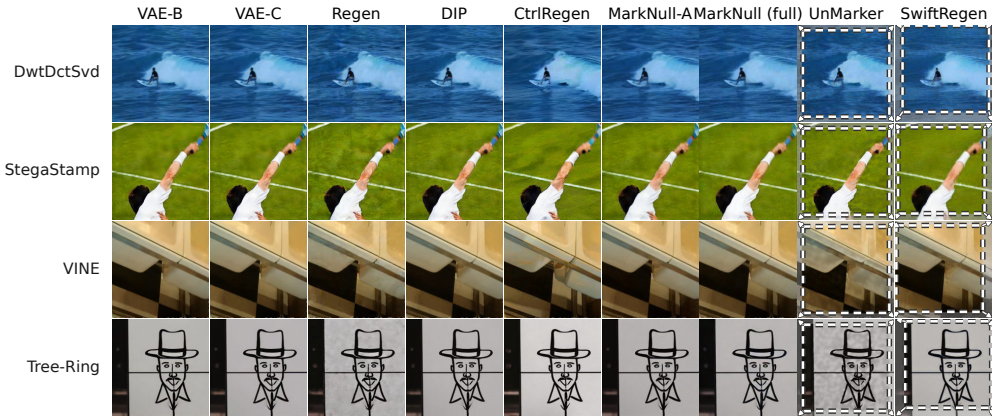


Figure 6: **Qualitative comparison across four watermarking schemes.** Each row uses the same watermarked source image across attacks. In the SwiftRegen column, the dashed rectangle marks the retained crop window. The output is shown inside the rectangle, and the image before cropping is faded outside.

similarity to the input. SwiftRegen has higher FID than the VAE and Regen baselines, showing that high semantic similarity can coexist with larger distributional differences.

Figure 6 compares visible changes across four watermarking schemes, and Figure 14 extends the comparison to all eleven. Aligned PSNR and SSIM for three schemes are reported separately in Table 11.

6 Analysis

6.1 Output Fidelity Does Not Guarantee Removal Transfer

We compare a full-image predictor trained to reproduce CtrlRegen outputs with the bounded residual predictor. On the same 100-image test subset, the full-image predictor achieves 22.71 dB PSNR relative to the regeneration output, compared with 22.34 dB for the residual predictor. Despite this higher output fidelity, it transfers watermark removal less effectively. Full-image prediction must reproduce both the image content and the update introduced by regeneration. The residual predictor retains the input through an identity connection and learns the update directly.

Table 3 evaluates both predictors on the same 100 images per watermarking scheme, using identical detector thresholds and no chromatic restoration or WCD. Residual prediction gives lower TPR on

Table 3: **Removal transfer with predictor outputs only.** TPR@1% FPR ↓ on the same 100 images per scheme, without chromatic restoration or WCD. Both predictors use fixed thresholds from the main benchmark, and all evaluated images are included.

	DwtDct	*DwtDctSvd*	*HiDDeN*	*RivaGAN*	*StegaStamp*	*Stable Sig.*	*VINE*	*RobustWide*	*GS*	*Tree-Ring*	*PRC*
Full-image prediction	0.02	0.41	0.31	0.98	1.00	1.00	1.00	1.00	1.00	0.97	0.99
Bounded residual prediction	0.03	0.03	0.00	0.29	0.01	0.12	0.80	0.13	1.00	0.26	0.17

nine schemes and ties on Gaussian Shading; full-image prediction is slightly better on DwtDct, at 0.02 versus 0.03. The largest improvements include StegaStamp (0.01 versus 1.00), RobustWide (0.13 versus 1.00), Stable Signature (0.12 versus 1.00), and HiDDeN (0.00 versus 0.31). These results establish that higher fidelity to the regeneration output does not guarantee stronger removal transfer. Learning bounded residual updates transfers removal more effectively in SwiftRegen.

6.2 Recoverability of Watermark Evidence

Watermark families differ substantially in evidence redundancy and recoverability. Interventions using clean references and refinement probes reveal that similar post-attack detection outcomes can conceal markedly different amounts of recoverable evidence (Appendix D).

Evidence survives, detection fails. On 100 images per scheme, WCD at $q = 0.90$ reduces detection from 100% to 14% for Gaussian Shading and to 0% for PRC. Restoring spatial correspondence using only the attacked image and the known crop geometry brings detection back to 100% for both schemes. Incorrect registration yields only 1% and 3%, respectively. The image recovery procedure uses neither original image content nor watermark information. Thus, evidence sufficient for detection survives WCD, while disrupted synchronization prevents its detection. Appendix E reports all three schemes and the unwatermarked controls. Separate pixel-replacement and refinement experiments examine other aspects of watermark recoverability (Appendix D).

7 Discussion and Conclusion

Implications. SwiftRegen shows that the removal behavior of iterative regeneration can be learned from clean image pairs and reused with a single forward pass. Output fidelity alone does not determine how well this behavior transfers. The WCD experiments identify a complementary condition for successful detection: surviving evidence must retain its correspondence with the detector's reference. Registration restores detection in the tested schemes, demonstrating that evidence persistence alone does not ensure reliable detection.

Limitations. Our evaluation uses one regeneration process and eleven watermarking schemes on COCO-derived images. Generalization to other regeneration models and image distributions remains to be evaluated. Residual changes and cropping introduce an image-quality tradeoff. The registration experiments use known crop geometry to examine the mechanism; they do not provide a blind recovery procedure.

Conclusion. SwiftRegen enables efficient watermark evasion by learning bounded updates from iterative regeneration and disrupting synchronization through WCD. The results show that faithfully reproducing regeneration outputs is insufficient for effective removal transfer, and that detection can fail even when watermark evidence survives. Together, these findings motivate evaluating watermark robustness in terms of both evidence persistence and synchronization.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks that require disclosure: implementing the research method and associated training, evaluation, and data-processing code; assisting with refinement of hypotheses and conceptual framing of failure mechanisms; interpreting reported experimental results; and assisting with translation. The core research design, methodology, and experimental plan were developed by the authors without generative AI. Generative AI tools were also used to assist with mathematical exposition and the presentation of theoretical derivations. We did not use generative AI tools to generate experimental data or synthetic data sets, or to support qualitative and thematic data analysis. All equations, derivations, assumptions, and resulting claims were independently checked and verified by the authors. Additionally, we used generative AI tools for tasks with recommended disclosure: searching for and summarizing relevant literature, suggesting manuscript structure, drafting and editing for readability, formatting references and LaTeX, and creating, revising, and refining scientific figures and figure labels. We reviewed all AI-assisted work, inspected and tested AI-generated code, independently reproduced the reported training and evaluation runs, and checked every numerical result, table, figure, equation, and citation against the original experiment outputs and primary sources. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Appendix A describes detector calibration, evaluation protocols, sample counts, and reference alignment. Appendix B provides the network architecture, training objectives and hyperparameters, regeneration configuration, post-processing operations, and runtime measurement protocol. Appendices C and D report additional ablations and diagnostic experiments. Appendix E describes the WCD registration experiments and measurements of representation changes, and Appendix F provides extended quality and qualitative results.

ETHICS STATEMENT

This work studies the robustness of invisible image watermarking to regeneration-based removal. Like other watermark removal methods, the technique is dual-use: learned rewriting and carrier desynchronization could be used to suppress provenance signals. Our goal is to characterize the computational cost and effectiveness of practical watermark removal attacks so that watermarking schemes can be designed and evaluated against them. All experiments use the public COCO2017 dataset and watermarked images produced with released watermarking implementations. We do not collect private data or involve human subjects. We report recovery experiments to distinguish detection failure from the disappearance of detectable watermark evidence.

REFERENCES

Ali Al-Haj. Combined DWT-DCT digital image watermarking. *Journal of Computer Science*, 3(9): 740–746, 2007.

Bang An, Mucong Ding, Tahseen Rabbani, Aakriti Agrawal, Yuancheng Xu, Chenghao Deng, Sicheng Zhu, Abdirisak Mohamed, Yuxin Wen, Tom Goldstein, and Furong Huang. WAVES: Benchmarking the robustness of image watermarks. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 1456–1492. PMLR, 2024.

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

Tu Bui, Shruti Agarwal, and John Collomosse. TrustMark: Robust watermarking and watermark removal for arbitrary resolution images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18629–18639, October 2025. doi: 10.1109/ICCV51701.2025.01731. URL https://openaccess.thecvf.com/content/ICCV2025/

html/Bui_TrustMark_Robust_Watermarking_and_Watermark_Removal_for_Arbitrary_Resolution_Images_ICCV_2025_paper.html.

Jie Cao, Qi Li, Zelin Zhang, Xiaodong Wu, Lingshuang Liu, Xiangman Li, and Jianbing Ni. MarkNull: Model-agnostic watermark removal in AI-generated images via on-manifold latent manipulation. In *35th USENIX Security Symposium (USENIX Security 26)*, pp. 4721–4740. USENIX Association, 2026. URL https://www.usenix.org/conference/usenixsecurity26/presentation/cao.

Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7939–7948, 2020.

Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2008.

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16 (8):2080–2095, 2007. doi: 10.1109/TIP.2007.901238.

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. doi: 10.1109/ICCV51070.2023.02053.

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

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017.

Runyi Hu, Jie Zhang, Ting Xu, Jiwei Li, and Tianwei Zhang. Robust-Wide: Robust watermarking against instruction-driven image editing. In *Computer Vision – ECCV 2024*, volume 15080 of *Lecture Notes in Computer Science*, pp. 20–37. Springer, 2024. doi: 10.1007/978-3-031-72670-5_2.

Zhaoyang Jia, Han Fang, and Weiming Zhang. MBRS: Enhancing robustness of DNN-based watermarking by mini-batch of real and simulated JPEG compression. In *ACM International Conference on Multimedia*, 2021.

Andre Kassis and Urs Hengartner. UnMarker: A universal attack on defensive image watermarking. In *IEEE Symposium on Security and Privacy*, pp. 2602–2620, 2025. doi: 10.1109/SP61157.2025.00005.

Chun Tong Lei, Hon Ming Yam, Zhongliang Guo, Yifei Qian, and Chun Pong Lau. Instant adversarial purification with adversarial consistency distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24331–24340, 2025. doi: 10.1109/CVPR52734.2025.02266.

Hengyue Liang, Taihui Li, and Ju Sun. A baseline method for removing invisible image watermarks using deep image prior. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=g85Vxlrq0O.

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 *European Conference on Computer Vision*, pp. 740–755, 2014. doi: 10.1007/978-3-319-10602-1_48.

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 *International Conference on Learning Representations*, 2025.

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

Nils Lukas, Abdulrahman Diaa, Lucas Fenaux, and Florian Kerschbaum. Leveraging optimization for adaptive attacks on image watermarks. In *International Conference on Learning Representations*, 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*, pp. 271–274. IEEE, 2008.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Tom Sander, Pierre Fernandez, Alain Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages. In *International Conference on Learning Representations*, 2025.

Fahad Shamshad, Nils Lukas, and Karthik Nandakumar. RAVEN: Erasing invisible watermarks via novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 82–91, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Shamshad_RAVEN_Erasing_Invisible_Watermarks_via_Novel_View_Synthesis_CVPR_2026_paper.html.

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.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9446–9454, 2018. doi: 10.1109/CVPR.2018.00984.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Advances in Neural Information Processing Systems*, volume 36, pp. 58047–58063, 2023. doi: 10.52202/075280-2529.

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*, pp. 12162–12171, 2024. doi: 10.1109/CVPR52733.2024.01156.

Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust invisible video watermarking with attention, 2019. URL https://arxiv.org/abs/1909.01285.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Wenliang Zhao, Lujia Bai, Yicheng Rao, Jie Zhou, and Jiwen Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

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 *Advances in Neural Information Processing Systems*, volume 37, pp. 8643–8672, 2024. doi: 10.52202/079017-0276.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. HiDDeN: Hiding data with deep networks. In *European Conference on Computer Vision*, pp. 657–672, 2018. doi: 10.1007/978-3-030-01267-0_40.

A Evaluation Protocols

A.1 TPR@1% FPR Calibration

For a multi-bit watermark of length n, let

$$B_j = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}[\hat{m}_{j,i} = m_{j,i}], \qquad \mathrm{BER}_j = 1 - B_j \tag{8}$$

denote the bit accuracy and BER of image j. Under the null hypothesis of independent random bit matches, $nB \sim \mathrm{Binomial}(n, 1/2)$. For a target false positive rate α, the multi-bit threshold is

$$k_{n,\alpha} = \min\left\{k : \sum_{i=k}^{n} \binom{n}{i} 2^{-n} \le \alpha\right\}, \tag{9}$$

and the empirical detection rate is

$$\widehat{\mathrm{TPR}}@\alpha\,\mathrm{FPR} = \frac{1}{N}\sum_{j=1}^{N} \mathbf{1}[nB_j \ge k_{n,\alpha}]. \tag{10}$$

The binomial calculation describes a decision rule for bit accuracy under the stated null model. In the reported evaluation, detector thresholds follow the calibration protocol for each scheme and remain fixed across attacks. For Gaussian Shading, tied scores on clean references give an achieved FPR of 0.90% at the nominal 1% operating point.

A.2 Fidelity and Aligned Evaluation

LPIPS is computed with AlexNet on images resized to 256×256, using the original watermarked input as the reference. For UnMarker only, the LPIPS reference is aligned using the attack's crop coordinates, output resolution, and interpolation. DINOv2 and CLIP report mean per-image cosine similarity, and FID compares feature distributions. These three metrics use the unaligned watermarked inputs for all methods. Table 11 separately reports PSNR and SSIM using references aligned with SwiftRegen's crop coordinates, resolution, and interpolation. DINOv2 uses the giant model and CLIP uses ViT-B/32; FID uses Inception-V3 features.

A.3 Sample Counts and Matched Randomness

The main benchmark uses 1,000 images per watermark, except for UnMarker and full MarkNull, which use 100 each. Full MarkNull uses the same images as its corresponding SwiftRegen comparison. MarkNull-A uses the released WRN denoiser with an SD1.5 proxy VAE. Its RobustWide entry uses a separate image set because the main set does not retain the per-image watermark messages. UnMarker follows its released configuration, including a 10% center crop before optimization.

The default WCD retains 90% of the height and width and restores the original resolution by bilinear interpolation. Matched comparisons reuse the same crop seed for each image. Figure 4 uses an independent subset of 1,000 images, whereas Table 1 reports the main benchmark.

For Table 3, both predictors use the same first 100 images per scheme and identical detector thresholds and decision directions. All 100 images produce valid scores. TPR is computed over the full set, without filtering by detection before attack.

B Method and Implementation Details

B.1 Network Architecture

The learned operator uses a residual U-Net with four scales and channel widths of 48, 96, 192, and 384. It has three downsampling operations, GroupNorm and SiLU activations, upsampling with transposed convolutions, and skip connections between corresponding scales. The final model has 7.92 million parameters and requires 116.64 GMAC for a 512×512 input. Table 4 lists the training settings, and Table 6 reports the computational and storage requirements.

Table 4: Two-stage training configuration.

Item	Pretraining	Refinement
Initialization	random	pretraining checkpoint
Epochs / steps	5 / 47,500	1 / 9,500
Learning rate	2×10^{-4}	5×10^{-5}
Residual bound s	0.5	0.8
Optimizer	AdamW	AdamW
Batch size	4	4

B.2 Training Objective

We organize the training objective into three groups,

$$\mathcal{L} = \mathcal{L}_{\text{operator}} + \mathcal{L}_{\text{structure}} + \mathcal{L}_{\text{distribution}}. \tag{11}$$

The operator loss supervises the predicted residual and output using the regeneration-induced update and the regeneration output:

$$\mathcal{L}_{\text{operator}} = \lambda_r \|r_\theta(x) - r^*(x)\|_1 + \lambda_x \|\mathcal{A}_\theta(x) - \mathcal{R}(x)\|_1. \tag{12}$$

The structure group combines perceptual, edge, and semantic feature matching, where the semantic term uses DINOv2 features (Oquab et al., 2024),

$$\mathcal{L}_{\text{structure}} = \lambda_{\text{lp}}\mathcal{L}_{\text{LPIPS}} + \lambda_{\text{edge}}\mathcal{L}_{\text{Sobel}} + \lambda_{\text{sem}}\mathcal{L}_{\text{DINO}}, \tag{13}$$

while the distribution group constrains residual energy, image- and residual-domain high-pass spectral statistics, and an adversarial loss applied to residual patches,

$$\mathcal{L}_{\text{distribution}} = \lambda_{\text{energy}}\mathcal{L}_{\text{energy}} + \lambda_{\text{fft}}^{\text{img}}\mathcal{L}_{\text{HP-FFT}}^{\text{img}} + \lambda_{\text{fft}}^{\text{res}}\mathcal{L}_{\text{HP-FFT}}^{\text{res}} + \lambda_{\text{adv}}\mathcal{L}_{\text{PatchAdv}}. \tag{14}$$

Let $\text{HP}(u) = |\mathcal{F}\{u\}| \odot M$ denote a spectral feature operation, where $\mathcal{F}$ is the orthonormal two-dimensional Fourier transform and M masks the central $25\% \times 25\%$ of the frequency plane. The two high-pass terms are

$$\begin{aligned} \mathcal{L}_{\text{HP-FFT}}^{\text{img}} &= \big\|\text{HP}\big(\mathcal{A}_\theta(x)\big) - \text{HP}\big(\mathcal{R}(x)\big)\big\|_1, \\ \mathcal{L}_{\text{HP-FFT}}^{\text{res}} &= \big\|\text{HP}\big(\log(1 + |\mathcal{F}\{r_\theta(x)\}|)\big) - \text{HP}\big(\log(1 + |\mathcal{F}\{r^*(x)\}|)\big)\big\|_1. \end{aligned} \tag{15}$$

The image term compares features of $\mathcal{A}_\theta(x)$ and $\mathcal{R}(x)$. In the residual term, HP is applied to the log-magnitude spectra shown in the equation. The residual energy term matches the magnitude and dispersion of the predicted residual to those of the regeneration-induced update,

$$\mathcal{L}_{\text{energy}} = \mathbb{E}_x\left[\left|\log\frac{\bar{r}_\theta(x) + \epsilon}{\bar{r}^*(x) + \epsilon}\right| + \left|\log\frac{\sigma(r_\theta(x)) + \epsilon}{\sigma(r^*(x)) + \epsilon}\right|\right], \tag{16}$$

where $\bar{r}(x)$ is the mean absolute residual over channels and pixels, $\sigma(\cdot)$ is the standard deviation over the same elements, and $\epsilon = 10^{-6}$. The semantic term $\mathcal{L}_{\text{DINO}}$ is the L1 distance between L2-normalized DINOv2 class-token features of the predicted output and of the clean input x, computed at 224×224 with the processor normalization; the model version is given in Appendix A.2. The active terms and weights differ between pretraining and refinement and are listed in Table 5. At inference, $\mathcal{A}_\theta$ requires one network evaluation and no diffusion sampling loop.

B.3 Training Hyperparameters

Training uses 40,000 COCO train2017 images in sorted order, split sequentially into 38,000 training pairs, 1,000 validation pairs, and 1,000 test pairs. Images are processed at 512×512 without random augmentation. Regenerated targets are produced with the released CtrlRegen configuration using Realistic Vision V4.0, a 50-step UniPC sampler (Zhao et al., 2023), image-to-image strength 0.5, and its spatial and semantic conditioning components. Table 5 lists the active loss terms and their weights in each stage. The high-pass spectral term acts on the restored image during pretraining and on the predicted residual during refinement; both stages use an L1 loss on Sobel magnitudes. The adversarial term is a non-saturating binary cross-entropy GAN on residual patches. The discriminator

Table 5: Active loss terms and weights per training stage. The high-pass spectral term is applied to the restored image at pretraining and to the residual at refinement; "–" marks a term that is not active in that stage.

Loss term	Pretraining	Refinement
Residual L1	1.0	1.0
Image L1	0.5	0.25
LPIPS (AlexNet)	0.1	0.08
Sobel edge	0.1	0.03
High-pass spectral (image)	0.02	–
High-pass spectral (residual)	–	0.08
Residual energy	–	0.12
Residual-domain Patch adversarial	–	0.03
Semantic feature matching (DINOv2)	0.05	0.03

maps a residual to a one-channel logit map and is trained with $\mathcal{L}_D = \frac{1}{2}[\mathrm{BCE}(D(r^*(x)), 1) + \mathrm{BCE}(D(\hat{r}_\theta(x)), 0)]$, where $\hat{r}_\theta(x)$ is detached from the predictor; the generator term is $\mathcal{L}_{\mathrm{PatchAdv}} = \mathrm{BCE}(D(r_\theta(x)), 1)$. The discriminator has four stride-2 4×4 convolutions with channel widths $32, 64, 128, 128$, GroupNorm and LeakyReLU(0.2) after the first three, and a final 3×3 convolution. It is optimized with AdamW at learning rate 10^{-4}, $(\beta_1, \beta_2) = (0.5, 0.999)$, weight decay 10^{-4}, and gradient clipping at 1.0, with a 1,000-step warmup. Regen uses Stable Diffusion 2.1 as its regeneration model.

B.4 Chromatic Restoration

Let $y = \mathcal{A}_\theta(x_w)$. Chromatic restoration uses local statistics of x_w to adjust y before WCD. Let $\mu_u(p)$ and $\sigma_u(p)$ denote the per-channel local RGB mean and standard deviation of image u at position p. Define

$$\begin{aligned} \rho(p) &= \mathrm{clip}_{[0.5,1.5]}\left(\frac{\sigma_{x_w}(p)}{\sigma_y(p) + \epsilon}\right), \\ y_{\mathrm{stat}}(p) &= \mu_{x_w}(p) + \rho(p)\big(y(p) - \mu_y(p)\big). \end{aligned} \tag{17}$$

The restoration is

$$\begin{aligned} y_1 &= (1-\beta)y + \beta y_{\mathrm{stat}}, \\ y_2 &= y_1 + \alpha\big(\mu_{x_w} - \mu_{y_1}\big), \\ \mathcal{C}(y; x_w) &= \mathrm{clip}\big(y_2 + \gamma\big[y_2 - G_\sigma * y_2\big], 0, 1\big), \end{aligned} \tag{18}$$

applied independently per RGB channel with reflection padding. The default values are a 31-pixel local window, $\alpha = 0.45$, $\beta = 0.25$, $\gamma = 0.15$, $\sigma = 1.0$, and $\epsilon = 10^{-6}$.

B.5 Spectral Effects of Carrier Re-indexing

Cropping and resizing change the image spectrum through windowing, translation, and rescaling. For an image y and crop window $m_{t,q}$, multiplication in image space gives

$$\mathcal{F}\{m_{t,q} y\} = (2\pi)^{-2}\,(M_{t,q} * Y), \tag{19}$$

where $Y = \mathcal{F}\{y\}$ and $M_{t,q} = \mathcal{F}\{m_{t,q}\}$. Using the Fourier convention $\mathcal{F}\{f\}(\omega) = \int f(p) e^{-i\omega^\top p}\, dp$, define the cropped, rescaled signal on $[0,1]^2$ and extend it by zero outside that domain. In the continuous idealization its spectrum is

$$Y_{t,q}(\omega) = \frac{1}{q^2} e^{i\omega^\top t/q} \mathcal{F}\{m_{t,q} y\}\left(\frac{\omega}{q}\right). \tag{20}$$

Windowing mixes spectral coefficients, translation changes phase, and resizing changes frequency coordinates. Bilinear interpolation introduces additional changes during discrete resampling. This derivation describes the image spectrum; effects on VAE latents and inversion outputs are measured in Appendix E.

Table 6: Deployment footprint of the learned operator.

Measurement	Value
Parameters	7.92 M
Compute at 512^2	116.64 GMAC
Disk, fp32 / INT8	30.2 / 12.1 MB
CPU, ONNX Runtime fp32	304 ms/image

B.6 Deployment and Footprint

Runtime is measured on an RTX 2080 Ti at batch size 1, including preprocessing and GPU synchronization. SwiftRegen timing includes the learned operator, chromatic restoration, and WCD. Disk I/O, PNG encoding, and detector calls are excluded. Table 6 reports model size, computation, storage, and CPU latency. Generating training pairs and training the operator are offline costs and are reported separately from per-image attack runtime. The regeneration cost is a property of the attack rather than of the watermark, so a single scheme-independent value is reported.

B.7 Full-Image and Residual Prediction

Both predictors use the same clean training images, cached CtrlRegen outputs, and data split described in Appendix B.3. They differ in architecture and optimization. The bounded residual predictor has 7.92 million parameters and is trained for $47{,}500$ steps at a learning rate of 2×10^{-4} with residual bound 0.5, followed by $9{,}500$ steps at 5×10^{-5} with bound 0.8.

The full-image predictor has 9.94 million parameters. It reuses ten of the thirteen pretrained operator modules, adds a bottleneck and filtered skip connections, and predicts the complete image through a sigmoid head without an identity path. It is trained for $2{,}000$ steps at a learning rate of 2×10^{-4} using losses for output matching, edges, Laplacian responses, equivariance, and LPIPS.

On a 32-image validation subset, PSNR relative to the clean input is 28.80 dB for full-image prediction and 22.90 dB for residual prediction. On a 100-image test subset, PSNR relative to the regeneration output is 22.71 dB and 22.34 dB, respectively. The 28.80 dB value therefore measures fidelity to the clean input, not to the regeneration output. The comparison evaluates two complete training configurations, including differences in architecture and initialization; the full-image predictor is initialized from the bounded residual model.

C Component and Training Ablations

C.1 Sensitivity to Training Seeds

We repeat the same two-stage training procedure with seeds 20270909 and 20270910 in addition to the original seed 2026. Each resulting model is evaluated on 1,000 images per scheme across seven schemes. Figure 7 and Table 7 report the variation across these training runs separately from the main benchmark.

C.2 Effect of Residual Strength

We vary the residual multiplier λ while keeping the trained predictor fixed: $\mathcal{A}_{\theta,\lambda}(x) = \mathrm{clip}(x + \lambda r_\theta(x), 0, 1)$. This diagnostic applies the learned operator alone, without chromatic restoration or WCD. Increasing residual strength generally improves removal at the cost of image fidelity, with different responses across schemes (Figure 8). DwtDctSvd and StegaStamp show sharp decreases in detection at intermediate strengths, while Tree-Ring remains detectable at higher strengths. This diagnostic uses each scheme's native detector threshold. The main evaluation uses a fixed attack configuration across schemes, consistent with the absence of target-detector feedback.

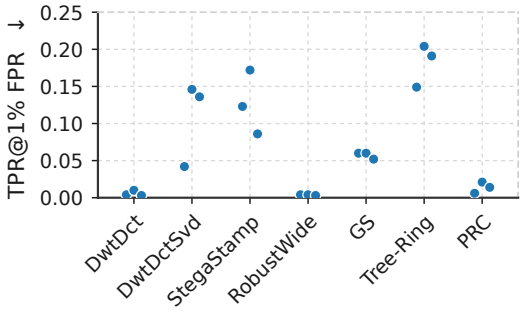


Figure 7: **Sensitivity to training seeds.** Each point reports full-pipeline TPR@1% FPR for one of three models trained with different random seeds, using 1,000 images per scheme.

Table 7: **TPR@1% FPR across three training runs.** Each scheme uses 1,000 images. S1, S2, and S3 correspond to training seeds 2026, 20270909, and 20270910.

Scheme	$\mathcal{A}_\theta$ only			$\mathcal{A}_\theta + \mathcal{C}$			Full		
	S1	S2	S3	S1	S2	S3	S1	S2	S3
DwtDct	0.006	0.024	0.010	0.007	0.023	0.014	0.004	0.010	0.003
DwtDctSvd	0.054	0.204	0.286	0.153	0.508	0.622	0.042	0.146	0.136
StegaStamp	0.012	0.011	0.002	0.159	0.240	0.078	0.123	0.172	0.086
RobustWide	0.128	0.560	0.294	0.172	0.649	0.362	0.004	0.004	0.003
Gaussian Shading	0.987	0.999	0.982	0.989	0.999	0.984	0.060	0.060	0.052
Tree-Ring	0.236	0.446	0.192	0.628	0.688	0.580	0.149	0.204	0.191
PRC	0.125	0.172	0.141	0.139	0.176	0.147	0.006	0.021	0.014

C.3 Training Variants

Table 8 compares each variant with its matched bounded residual baseline on ten held-out images per scheme. Each variant and its baseline share the same subset; different rows can use different subsets. All evaluated images are included, without filtering by detection before attack.

The structure-anchor variant adds a constraint on low-frequency structure, and the residual-direction variant constrains the direction of the predicted update. Continued training resumes the bounded residual model with its original configuration. Best-of-K target selection chooses a regenerated target for each training sample. The final variant applies a separate refinement network after the residual predictor. The two constraint variants use test$[40{:}50]$, continued training uses test$[50{:}60]$, target selection uses test$[60{:}70]$, and the refinement configuration uses test$[80{:}90]$.

None of these variants improves both evaluated schemes consistently. Continued training and target selection reduce StegaStamp detection but increase RobustWide detection, while the two constraint variants worsen both. Refinement increases StegaStamp detection from 0.10 to 1.00 and leaves RobustWide unchanged. This recovery indicates that evidence sufficient for StegaStamp detection remains after the learned operator. The full-image comparison is reported separately in Table 3.

C.4 Additional WCD Configurations

Table 9 reports additional configurations: $\mathcal{A}_\theta$ and $\mathcal{A}_\theta + \mathcal{C}$ without WCD, centered cropping with $q = 0.94$ on 100 images, and the default random cropping with $q = 0.90$ under two training seeds. The centered run uses Lanczos interpolation, whereas the default configuration uses bilinear interpolation. These configurations differ in sampling and preprocessing as well as retained fraction.

C.5 Additional WCD Mechanism Controls

We compare centered cropping and resizing with JPEG compression, Gaussian blur, and additive Gaussian noise on Gaussian Shading, Tree-Ring, and PRC. Each condition uses 50 images and the same

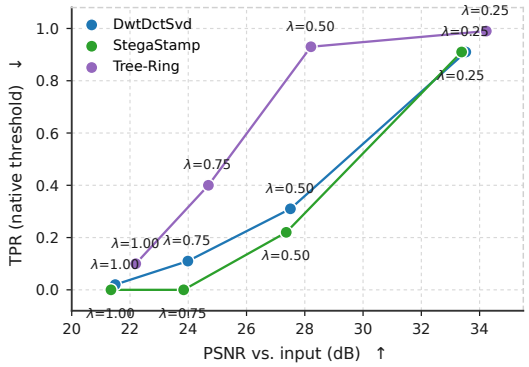


Figure 8: **Effect of residual strength on detection and image quality.** Detection rate versus PSNR relative to the input for DwtDctSvd, StegaStamp, and Tree-Ring, using 100 images per scheme. Each point corresponds to one value of the residual multiplier λ (Appendix C); the identity setting has infinite PSNR and is omitted. This diagnostic uses each scheme's native detector threshold, separately from the calibrated main benchmark.

Table 8: **Comparison of training variants.** Each variant is compared with its matched bounded residual baseline on ten held-out images per scheme. Subsets can differ across rows. Lower TPR is better. The full-image comparison is reported in Table 3.

Variant	StegaStamp		RobustWide	
	res.	variant	res.	variant
Structure anchor	0.20	0.30	0.00	0.70
Residual direction	0.20	0.50	0.00	0.90
Continued training	0.10	0.00	0.40	0.70
Best-of-K target	0.20	0.00	0.30	0.50
Residual→refiner	0.10	1.00	0.00	0.00

fixed detector thresholds. Cropping uses retained side fractions q in $\{0.98, 0.96, 0.94, 0.92, 0.90\}$ with Lanczos resampling. JPEG quality is 40 or 70; the other control parameters follow the settings reported with the experiment. The crop and other distortion controls use the same image manifest, but their perceptual distortion levels are not matched.

Gaussian Shading remains fully detectable under the reported non-geometric controls but falls to 0.14 after cropping with $q = 0.90$. Tree-Ring retains detection rates of 0.92–0.98 under cropping but falls to 0.04 under blur with radius 4. In the additional blur control, Tree-Ring detection is 0.00 at radii 6 and 8. These responses show that the schemes respond differently to geometric and other image transformations. The registration experiment in Appendix E directly tests the role of spatial correspondence.

A separate experiment applies centered WCD after CtrlRegen on 100 images. For $q = 0.94$ and $q = 0.90$, detection rates are 0.09 and 0.05 for Gaussian Shading, 0.02 and 0.00 for PRC, and 0.15 and 0.23 for Tree-Ring. This experiment uses CtrlRegen followed by WCD and is separate from the matched SwiftRegen component evaluation in Figure 4.

C.6 Recovery After Bounded Blind Re-registration

A separate $n = 50$ experiment probes whether a bounded *blind* scale/translation search can find the right geometry. For a known center crop-resize, a small scale/translation grid is searched and the original detector re-run, comparing against the oracle inverse transform. Re-registration raises TPR@1% FPR from 0.26 to 0.78 for Gaussian Shading and from 0.26 to 0.50 for PRC, indicating that much of their loss is reversible by coordinate re-registration. For RobustWide, detection remains at 0.04 after the search and under the tested oracle inverse transform, showing that these registration operations do not recover detection in this experiment. This protocol is distinct from the known-

Table 9: **Additional WCD configurations.** TPR@1% FPR under the reported settings. The two no-WCD columns report the $\mathcal{A}_\theta$-only and $\mathcal{A}_\theta$+$\mathcal{C}$ stages of the seed-20270909 checkpoint (1,000 images per scheme). The centered column is a 100-image center-crop run (retained side fraction $q = 0.94$, Lanczos). The last two columns report the full pipeline ($\mathcal{A}_\theta$+$\mathcal{C}$ + WCD, random crop $q = 0.90$, bilinear) under training seeds 2026 and 20270909. Lower is better.

	no WCD (seed 20270909)		6% (n=100)	10% (two seeds)	
Method	$\mathcal{A}_\theta$	+$\mathcal{C}$	center crop	2026	20270909
DwtDct	0.024	0.023	0.01	0.004	0.010
PRC	0.172	0.176	0.01	0.006	0.021
DwtDctSvd	0.204	0.508	0.12	0.042	0.146
RobustWide	0.560	0.649	0.03	0.004	0.004
StegaStamp	0.011	0.240	0.35	0.123	0.172
Gaussian Shading	0.999	0.999	0.34	0.060	0.060
Tree-Ring	0.446	0.688	0.36	0.149	0.204

geometry registration diagnostic of Appendix E: it asks whether the geometry can be *found* blindly, with different schemes and sample size, and the two are not pooled.

D Removal Diagnostics

D.1 Pixel Replacement Using Clean References

We first examine how detection responds when selected pixels are restored to their unwatermarked values. For paired images x and x_w, define the embedding residual $e = x_w - x$. We rank pixels by the mean absolute residual across RGB channels, $a_p = \frac{1}{3}\sum_c |e_{p,c}|$, and replace the top $K\%$ with their values from x:

$$x_p^{(K)} = \begin{cases} x_p, & p \in \Omega_K, \\ (x_w)_p, & p \notin \Omega_K. \end{cases} \tag{21}$$

The curves report detection rates for DwtDct, DwtDctSvd, RobustWide, and StegaStamp, and mean bit accuracy for HiDDeN, MBRS (Jia et al., 2021), RivaGAN, TrustMark, and WAM (Sander et al., 2025) (Figure 9). For DwtDct, we use the 71 of 100 images detected before pixel replacement. The resulting curves measure sensitivity to replacing pixels in this amplitude order and show differences in how watermark information is distributed across the image.

D.2 Recoverability After Refinement

The pixel-replacement experiment examines the original embedded signal. We next test whether watermark information remains recoverable after the learned operator. A separate residual U-Net is trained on clean images to map $\mathcal{A}_\theta(x)$ back to x, using an image L1 loss and LPIPS. At inference, this refinement network receives only the operator output and uses no clean counterpart or watermark information. It performs no cropping or resizing and is not part of the SwiftRegen attack pipeline.

Table 10 and Figure 10 compare bit accuracy before the attack, after the learned operator, and after refinement. Bit accuracy increases after refinement for most evaluated schemes. RobustWide remains low, while WAM retains high bit accuracy throughout. These results show that similar outcomes after the learned operator can be followed by different levels of recovery.

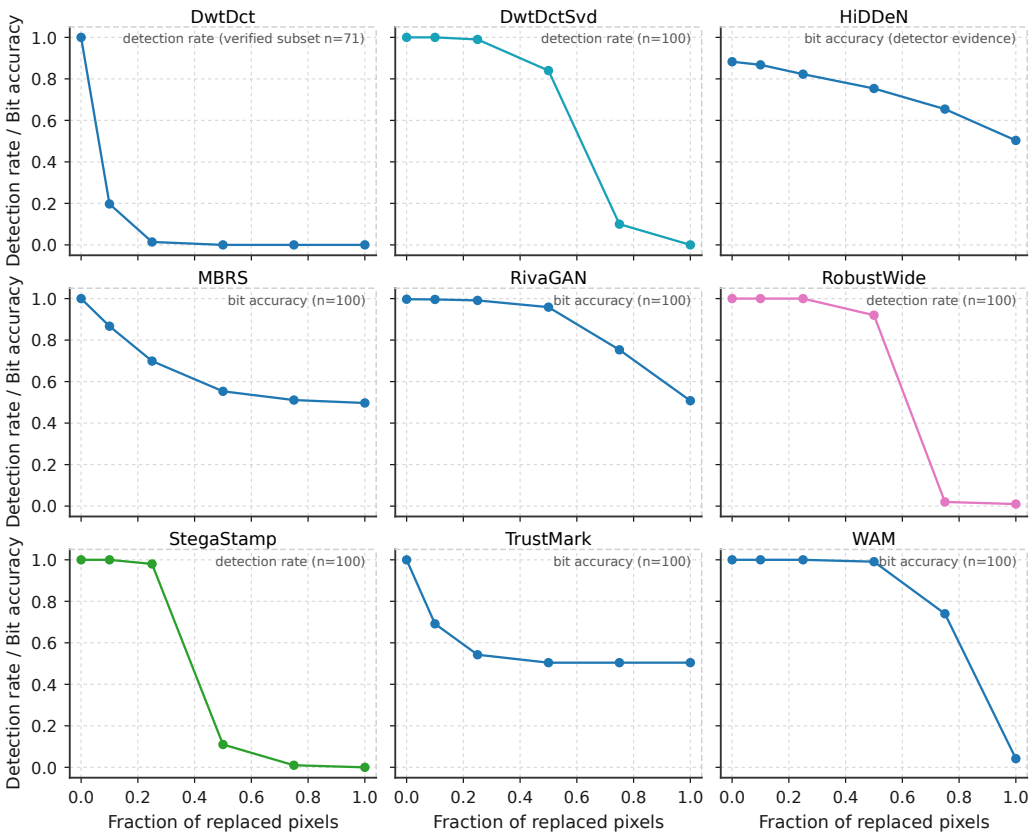


Figure 9: **Detection after replacing pixels with clean reference values.** Pixels are selected in descending order of embedding residual magnitude and replaced with the corresponding values of the paired unwatermarked image; the horizontal axis is the fraction of replaced pixels. Each panel reports detection rate or bit accuracy for one scheme, as labeled in the panel. DwtDct uses the 71 images detected before replacement; the other panels use 100 images.

Table 10: **Refinement probe across nine post-hoc schemes.** Bit accuracy before the attack, after the learned operator, and after refinement ($n = 20$; StegaStamp and RobustWide $n = 10$). Lower is not better here; the quantity is accuracy.

Method	watermarked	learned operator	refined
WAM	1.000	1.000	0.998
MBRS	0.999	0.448	0.882
DwtDctSvd	1.000	0.413	0.663
StegaStamp	0.999	0.531	0.886
HiDDeN	0.873	0.612	0.765
RivaGAN	0.986	0.673	0.805
TrustMark	1.000	0.582	0.683
DwtDct	0.972	0.567	0.625
RobustWide	1.000	0.388	0.303

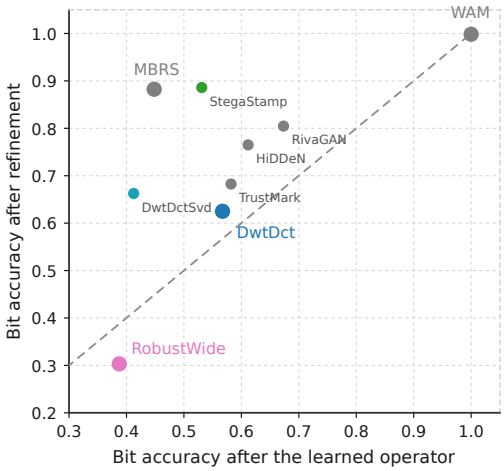


Figure 10: **Watermark recovery after refinement.** Each point compares bit accuracy after the learned operator with bit accuracy after a separate refinement network. The diagonal indicates no change. Most schemes show increased bit accuracy, RobustWide remains low, and WAM retains high accuracy throughout.

E Evidence Recoverability and Cross-Domain Effects of WCD

E.1 Detection Recovery Through Known-Geometry Registration

Recovery through spatial registration. We evaluate Tree-Ring, Gaussian Shading, and PRC using 100 watermarked images and 100 unwatermarked controls per scheme. WCD retains a side fraction $q = 0.90$. Recovery uses only the attacked image, the original canvas size, and the crop box. We resize the attacked image back to the crop dimensions with bilinear interpolation, place it on the original canvas, and fill the remaining area by replicating the nearest boundary values. Correct and incorrect registration use the same resized patch and differ only in placement. Incorrect positions are fixed before evaluation and are not selected using detector scores. All conditions use the original detector thresholds. For a horizontal placement range of 0 through M_x, the incorrect position is $l_{\text{wrong}} = (l + \lceil (M_x + 1)/2 \rceil) \bmod (M_x + 1)$, with the same rule applied vertically; a zero placement range remains unchanged. This experiment uses known crop geometry to test evidence recoverability; it does not evaluate blind search, and the observed FPRs are empirical estimates from 100 controls.

Correct registration restores detection to 100% for all three schemes, compared with 92%, 14%, and 0% after WCD for Tree-Ring, Gaussian Shading, and PRC, respectively (Figure 11). Among images detected originally but missed after WCD, correct registration recovers $8/8$, $86/86$, and $100/100$; incorrect registration recovers $3/8$, $0/86$, and $3/100$. The corresponding overall detection rates after incorrect registration are 64%, 1%, and 3%. On unwatermarked controls, correct registration gives empirical FPRs of 1%, 0%, and 1%; PRC reaches 4% under incorrect registration. The figure reports Wilson intervals for all conditions. These results show that detection can be restored by correcting spatial correspondence without supplying original image content or watermark information. As supporting evidence, Gaussian Shading message bit accuracy returns to 1.000 after correct registration.

Comparison with the earlier search experiment. A separate experiment on 50 images per scheme searched a bounded grid of scale and translation parameters after centered cropping and resizing. Detection increased from 0.26 to 0.78 for Gaussian Shading and from 0.26 to 0.50 for PRC. RobustWide remained at 0.04, including under the tested oracle inverse transform. These results describe recovery under that search protocol, whereas the experiment above restores the known geometry on a different 100-image evaluation set. Failure to recover RobustWide with the tested operations does not establish that all embedded evidence was destroyed.

E.2 Effects on VAE Latents and Recovered Noise

Effects on encoded representations. We compare the original image, a resize-only control, and WCD with retained side fractions of 0.95 and 0.90 on the same 100 images per scheme. The resize-only control reduces the full image to 90% of each spatial dimension and restores its original size. We measure changes in the image, its VAE encoding, and the noise recovered using the fixed inversion configuration of each scheme, using deterministic posterior encoding and no VAE decoding step. WCD uses the released random crop-and-resize with crop size $\mathrm{round}(w\,q)$ and a shared per-image position for the two strengths; every condition is re-encoded and re-inverted from its own latent, never reusing the original intermediate states. Coordinate correlation is computed after centering each channel and is averaged across channels. For the phase comparison, we use the orthonormal two-dimensional Fourier transform of each centered channel and exclude the DC component, weighting phase differences by the product of the two spectral magnitudes:

$$D_\phi(A, B) = \frac{\sum_{k \in V} |A_k||B_k| \left[1 - \cos(\arg B_k - \arg A_k)\right]}{2 \sum_{k \in V} |A_k||B_k| + \epsilon}. \tag{22}$$

Here V contains the valid non-DC coefficients whose magnitudes in both representations exceed 10^{-6} times the RMS magnitude of the original spectrum; the figure reports mean and standard deviation of per-image metrics over 100 images.

Figure 12 reports coordinate correlation and weighted phase discrepancy relative to each image's original representation. WCD changes all three representations. For Gaussian Shading at $q = 0.90$, mean coordinate correlation is 0.512 in image space, 0.335 in the VAE latent, and 0.023 in the recovered noise; the corresponding phase discrepancies are 0.213, 0.298, and 0.486. These values

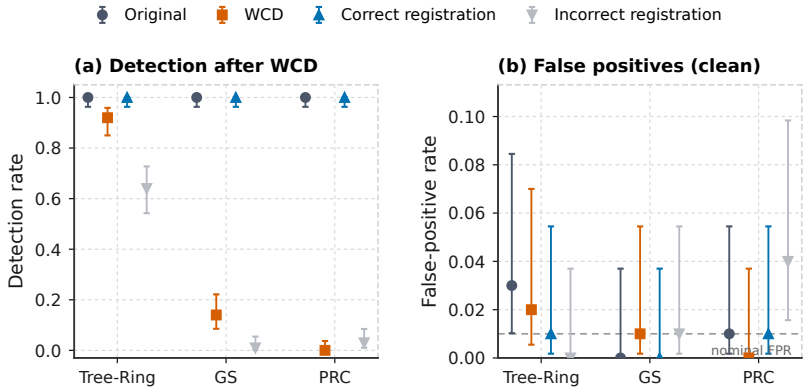


Figure 11: **Detection recovery through spatial registration.** (a) Detection rates before WCD, after WCD ($q = 0.90$), and after correct or incorrect registration. Registration uses only the attacked image and crop geometry. (b) Empirical false-positive rates on unwatermarked controls under the corresponding transformations; the horizontal line marks the nominal 1% FPR. Original detector thresholds remain fixed. Each scheme uses 100 watermarked images and 100 unwatermarked controls. Error bars denote 95% Wilson intervals.

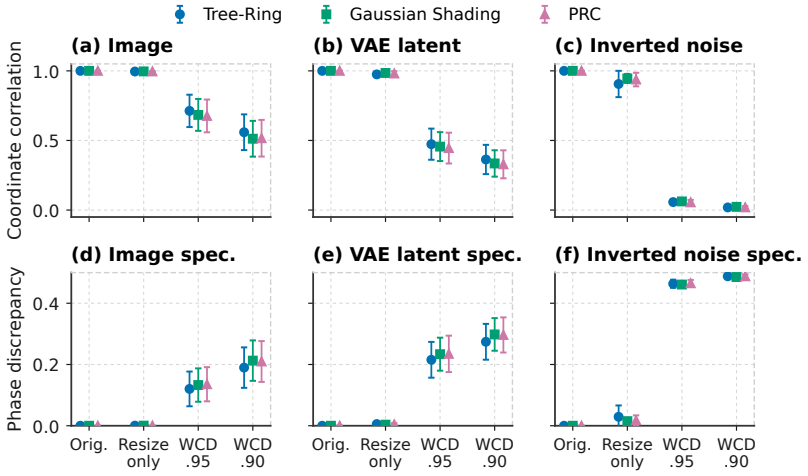


Figure 12: **Cross-domain effects of WCD.** Coordinate correlation and weighted phase discrepancy relative to the original watermarked input are measured in image space, VAE-encoded latents, and noise recovered with the fixed inversion configuration of each scheme. Conditions include the original image, the resize-only control, and WCD at $q = 0.95$ and $q = 0.90$. Results show mean $\pm$ standard deviation over 100 images per scheme.

describe correspondence within each representation and are not a common scale of watermark damage. Resize-only leaves detection at 100% for all three schemes. At $q = 0.95$, detection is 98% for Tree-Ring, 74% for Gaussian Shading, and 1% for PRC; the $q = 0.90$ results appear in Figure 11. Combined with the registration results, the representation changes show that substantial misalignment can coexist with recoverable watermark evidence. Figure 13 visualizes these changes in the image residual, recovered initial noise, and its Fourier representation for RobustWide, Gaussian Shading, and Tree-Ring, respectively.

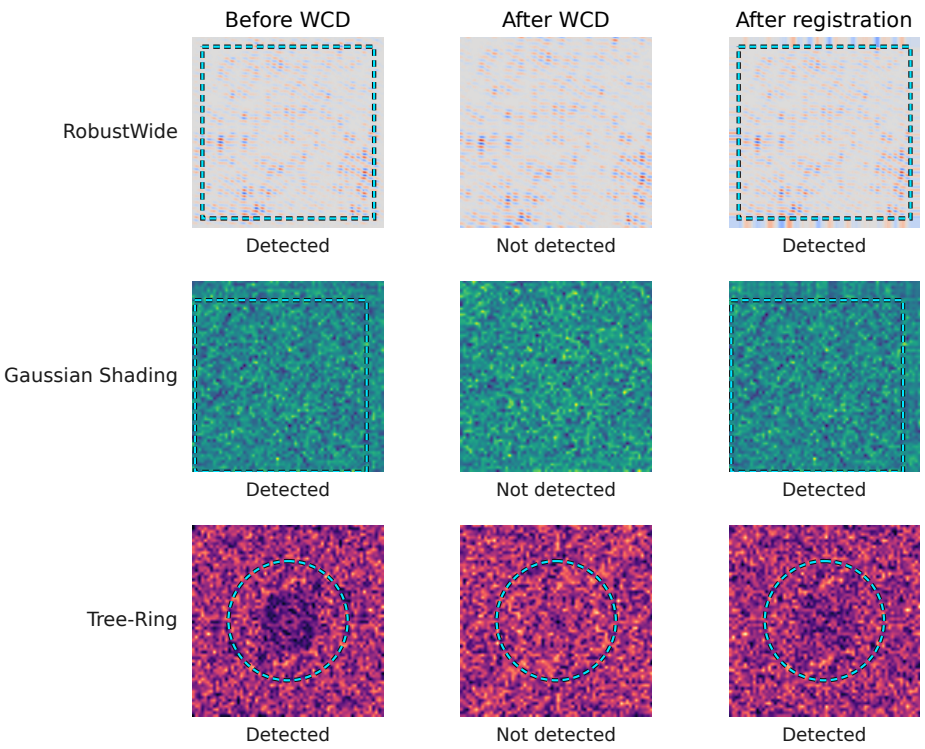


Figure 13: **Watermark representations before WCD, after WCD, and after registration using the known crop geometry.** Rows show paired image residuals for RobustWide (signed $x_w - x$, shared scale), recovered initial noise for Gaussian Shading (channel 0), and Fourier spectra of recovered initial noise for Tree-Ring ($\mathrm{fftshift}(\log(1 + |\mathcal{F}\{z\}|))$ of channel 0, with a thin circle marking the true watermark band). Each row uses the same sample and display scale across all three conditions. For RobustWide, identical transformations are applied to the watermarked image and its unwatermarked counterpart before subtraction. For Gaussian Shading and Tree-Ring, all representations are extracted using the same inversion procedure. Detection labels report the original detector's decision at the fixed threshold.

F Extended Quality and Qualitative Results

F.1 Image Fidelity After Geometric Alignment

Table 11: Aligned fidelity for Gaussian Shading, Tree-Ring, and PRC under the complete SwiftRegen pipeline.

Scheme	PSNR ↑	SSIM ↑
Gaussian Shading	23.24	0.630
Tree-Ring	23.82	0.616
PRC	23.89	0.680

F.2 Full Qualitative Comparison

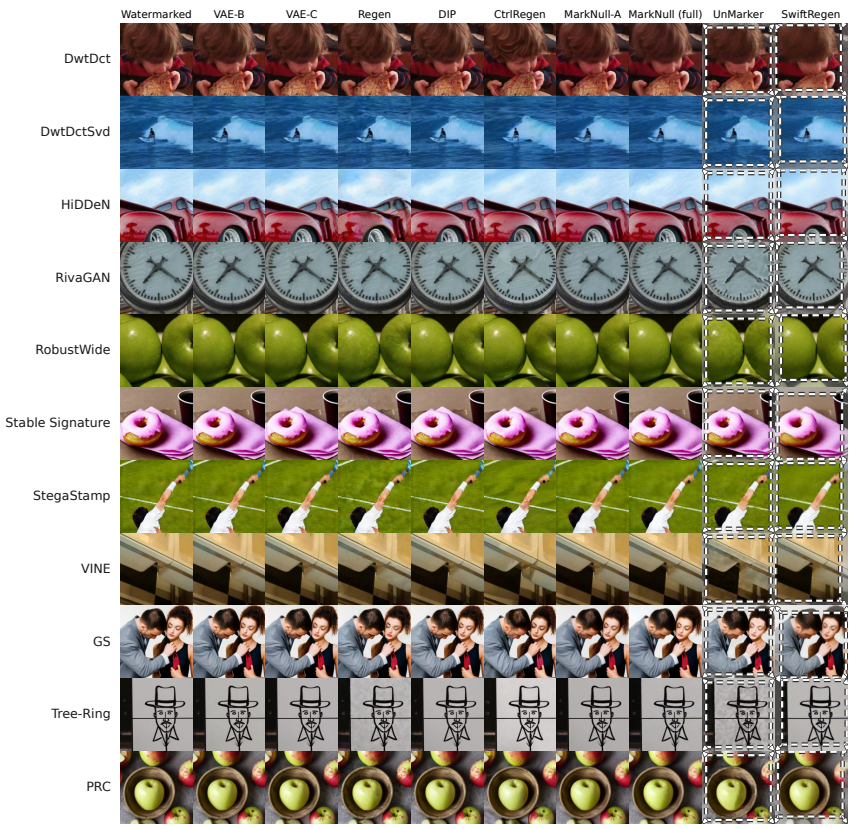


Figure 14: **Qualitative comparison across eleven watermarking schemes.** Each row uses the same watermarked source image. In the SwiftRegen column, the output is shown inside the retained crop window, while the image before cropping is faded outside.