PREFIX: ROBUST INVERSION-BASED IMAGE WATERMARKING VIA GENERATION-SIDE PRE-COMPENSATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Robustness to distortions is essential for reliable generative image watermarking. Among existing approaches, inversion-based methods offer a promising solution by embedding watermarks into the initial noise latent and recovering them through inversion. However, distortions can perturb the inversion input and compromise watermark recovery. To address this, we take a generation-side perspective on robustness, accounting for distortion effects on watermark extraction before image release. Building on this insight, we propose PreFix, a generation-side pre-compensation framework that optimizes an imperceptible residual per image to improve robustness without modifying the underlying inversion algorithm. Since non-geometric degradations perturb image statistics while geometric transformations introduce spatial misalignment, PreFix combines two complementary mechanisms within a unified residual optimization. Specifically, (1) degradation-pool optimization improves watermark recovery under sampled non-geometric degradations; and (2) alignment-aware optimization uses a lightweight alignment predictor to guide the residual to support transformation estimation and pre-inversion correction. Extensive experiments across watermarking methods and generative models demonstrate substantial robustness gains under both distortion types.

1 INTRODUCTION

Recent advances in diffusion models Ho et al. (2020); Rombach et al. (2022) have enabled the synthesis of highly realistic images, raising concerns about their potential misuse for spreading disinformation. These concerns highlight the need for reliable provenance verification and source traceability of generated content. Generative image watermarking offers a promising solution by embedding imperceptible signals into generated images. Its practical effectiveness, however, depends on robustness: the embedded watermark should remain detectable and recoverable after the image undergoes distortions during distribution and use.

Among existing generative image watermarking methods, inversion-based frameworks have emerged as an effective approach. As illustrated in Fig. 1, they embed watermarks into the initial noise latent before image generation and recover this latent through inversion for watermark extraction, commonly using Denoising Diffusion Implicit Models (DDIM) inversion Song et al. (2020). Representative methods improve robustness through their embedding designs: Tree-Ring Wen et al. (2023) imposes structured patterns in the frequency domain, while Gaussian Shading Yang et al. (2024) introduces spatial redundancy into the watermark encoding. Although these designs provide robustness to certain perturbations, reliable extraction still depends on preserving watermark information during latent recovery. Image distortions can disrupt this process by perturbing the inversion input, thereby compromising watermark detection and decoding.

This dependence on inversion motivates a generation-side perspective on robustness. Before releasing a watermarked image, the generator has access to both the image and the target watermark, allowing extraction to be evaluated under sampled distortions. The resulting feedback can guide image optimization to improve subsequent watermark recovery. We therefore ask: *Can a small, imperceptible residual added before image release improve watermark extraction under subsequent distortions without modifying the underlying inversion algorithm?*

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.

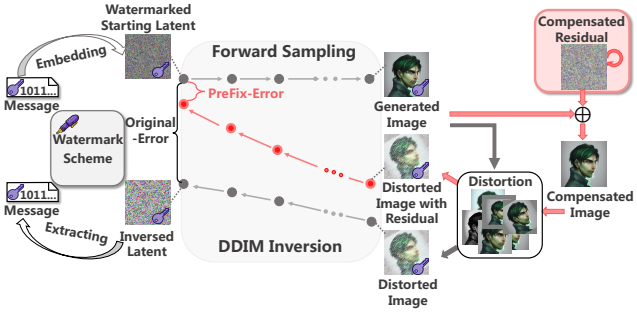


Figure 1: Overview of inversion-based generative image watermarking and PreFix.

Building on this insight, we propose **PreFix**, a generation-side pre-compensation framework that optimizes an imperceptible residual for each generated watermarked image to improve robustness under distortions. PreFix accounts for two distinct effects of image distortions: non-geometric degradations perturb image statistics, while geometric transformations introduce spatial misalignment. To address both, PreFix combines two complementary mechanisms within a unified residual optimization: (1) degradation-pool optimization improves watermark recovery under sampled non-geometric degradations; and (2) alignment-aware optimization uses a lightweight alignment predictor to guide the residual toward supporting transformation estimation and pre-inversion correction. Together, these mechanisms improve robustness while preserving visual quality.

In summary, we make the following contributions:

- We propose PreFix, a generation-side pre-compensation framework that improves robustness of inversion-based watermarking through per-image residual optimization before release.
- We develop a unified residual optimization that combines degradation-pool optimization for non-geometric distortions with alignment-aware optimization for geometric distortions.
- Extensive experiments across multiple watermarking methods show that PreFix substantially improves robustness under both non-geometric and geometric distortions.

2 Related Work

2.1 Diffusion Models

Diffusion models Ho et al. (2020) generate images by progressively denoising samples from a simple prior, typically a Gaussian distribution, and have achieved strong performance in image synthesis Dhariwal & Nichol (2021), editing Kawar et al. (2023), and inpainting Lugmayr et al. (2022). Latent diffusion models (LDMs) Rombach et al. (2022), such as Stable Diffusion, improve computational efficiency by operating in the latent space of a variational autoencoder (VAE). DDIM Song et al. (2020) enables deterministic sampling with fewer denoising steps. Importantly, this deterministic formulation also supports approximate inversion to recover the initial noise latent from an image, enabling applications such as inversion-based watermark extraction.

2.2 Generative Image Watermarking

Generative image watermarking for diffusion models embeds messages into generated images to enable detection and traceability. Existing methods fall into three categories. ***1). Post-hoc Methods*** Cox et al. (2008); Zhang et al. (2019); Zhu et al. (2018); Jia et al. (2021); Ma et al. (2022); Bui et al. (2025) apply image watermarking directly to diffusion-generated images. ***2). Fine-tuning-based Methods*** adapt components of LDMs or introduce trainable watermarking modules, typically within the VAE decoder: Stable Signature Fernandez et al. (2023) fine-tunes the decoder with supervision from a pre-trained extractor, while LaWa Rezaei et al. (2024) inserts coarse-to-fine watermarking modules into frozen decoder layers to improve capacity and robustness. ***3). Inversion-based Methods*** embed watermarks into the initial latent prior to sampling and recover them by inverting the generated image back to the starting latent. Tree-Ring Wen et al. (2023) introduces structured patterns into the latent Fourier domain for watermark detection, while RingID Ci et al. (2024) extends this line of

work to improve multi-key identification. Gaussian Shading (GS) Yang et al. (2024) embeds multi-bit messages while preserving the Gaussian distribution of the initial latent. PRC Gunn et al. (2025) constructs watermarked initial latents using pseudorandom error-correcting codes.

3 METHOD

3.1 PROBLEM FORMULATION

The primary objective of our method is to optimize a small, imperceptible generation-side residual r on each watermarked image to improve extraction robustness under distortions.

Basic Formulation. We consider an inversion-based watermarking pipeline where a message $m \in \{0,1\}^L$ is embedded into an initial latent z via an embedding mechanism $E(\cdot)$, i.e., $z = E(m)$. Given the text condition C, the generation pipeline $G(\cdot;\cdot)$ produces a watermarked image $x = G(z; C)$. Since the original text condition is typically unavailable at extraction time, we apply inversion $I(\cdot;\cdot)$ under the $\emptyset$-condition, typically using DDIM, to recover $\hat{z} = I(x; \emptyset)$. The corresponding extraction mechanism $D(\cdot)$ then produces the recovered message $\hat{m} = D(\hat{z})$, with the goal of recovering the embedded message, i.e., $\hat{m} = m$.

In practice, the released image may undergo a distortion $T(\cdot)$ before extraction. Distortions perturb the inversion input and can impair latent recovery, potentially yielding:

$$D\big(I(T(x); \emptyset)\big) \neq m. \tag{1}$$

Pre-compensation Objective. To improve watermark recovery under distortions, we optimize a generation-side residual r and release the compensated image $x + r$. Let $\mathcal{P}$ denote the distortion distribution. The basic pre-compensation objective minimizes the expected watermark extraction loss subject to an image fidelity constraint:

$$\min_{r} \ \mathbb{E}_{T\sim\mathcal{P}}\Big[\mathcal{L}_{\mathrm{wm}}\big(D(I(T(x+r); \emptyset)), m\big)\Big] \quad \text{s.t.} \quad \mathcal{L}_{\mathrm{img}}(x+r, x) \leq \epsilon. \tag{2}$$

where $\mathcal{L}_{\mathrm{wm}}$ denotes the watermark extraction loss, $\mathcal{L}_{\mathrm{img}}$ encourages imperceptibility of the residual, and ϵ specifies the fidelity budget.

3.2 PREFIX

To realize the pre-compensation objective, we consider two distortion families with different effects on inversion: non-geometric distortions perturb image statistics, while geometric distortions additionally induce spatial misalignment. For non-geometric distortions, we optimize the residual over a pool of common degradations to improve watermark recovery. For geometric distortions, we introduce pre-inversion alignment to correct spatial misalignment. To support geometric correction without access to the original image, we jointly learn a universal anchor template e and an alignment predictor A for transformation estimation. The template also serves as the initialization for the per-image residual. These designs form the two complementary mechanisms of PreFix: degradation-pool optimization and alignment-aware optimization.

As illustrated in Fig. 2, PreFix comprises three stages. First, we jointly train the anchor template e and alignment predictor A. Second, we initialize $r^{(0)} = e$ and optimize the residual under sampled distortions to obtain the compensated image. Finally, the potentially distorted image is aligned and inverted for watermark extraction.

3.2.1 TRAINING OF THE ALIGNMENT PREDICTOR $A(\cdot)$

The alignment predictor $A(\cdot)$ is trained to predict anchor coordinates from geometrically distorted images for subsequent correction. To facilitate anchor prediction, we introduce a universal anchor template e shared across images. Given a host image x^h, we add the template to obtain $x^e = x^h + e$ and apply a sampled geometric distortion through the geometric noise layer $\mathcal{N}_G$:

$$x^g = \mathcal{N}_G(x^e; \xi), \tag{3}$$

where ξ denotes the sampled transformation parameters.

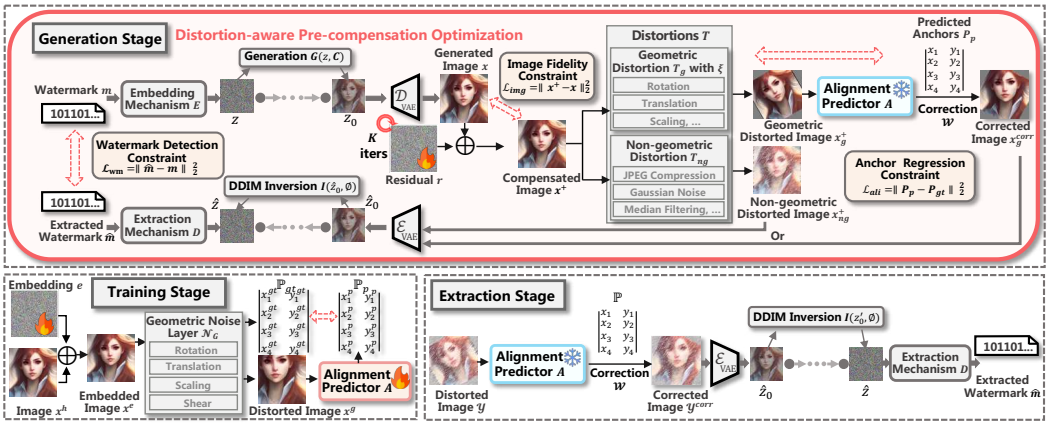


Figure 2: Overview of PreFix, including alignment predictor training, generation-side residual optimization, and watermark extraction.

Given x^g, the predictor estimates the anchor coordinates as $P_p = A(x^g)$. The ground-truth coordinates P_{gt} are obtained by applying the sampled transformation ξ to a predefined set of reference anchors. We jointly optimize the alignment predictor $A(\cdot)$ and the template e by minimizing a keypoint regression loss:

$$\mathcal{L}_{\text{pre}} = \|P_p - P_{\text{gt}}\|_2^2 = \left\|A(x^g) - P_{\text{gt}}\right\|_2^2. \tag{4}$$

After pre-training, the predictor is fixed for subsequent residual optimization and watermark extraction. Architectural details of $A(\cdot)$ are provided in the Appendix.

3.2.2 Distortion-aware Pre-compensation Optimization

Given a generated watermarked image x, we initialize the residual as $r^{(0)} = e$ and optimize it for K iterations to improve watermark recovery under distortions. Only r is updated, while the alignment predictor and the underlying watermarking pipeline remain fixed. At each iteration, we form the compensated image $x^+ = x + r$ and sample a distortion $T \sim \mathcal{P}$, which is either a non-geometric degradation T_{ng} or a geometric transformation $T_{\text{g}}(\cdot;\xi)$. The distortion-free case follows the non-geometric branch with an identity transformation.

Degradation-pool Optimization. For a sampled non-geometric distortion, we obtain $x_{\text{ng}}^+ = T_{\text{ng}}(x^+)$ and evaluate watermark recovery by inverting and decoding x_{ng}^+. The watermark extraction loss guides the residual to improve robustness across the sampled degradations.

Alignment-aware Optimization. For a sampled geometric distortion with parameters ξ, we obtain $x_{\text{g}}^+ = T_{\text{g}}(x^+;\xi)$ and predict the anchor coordinates $P_p = A(x_{\text{g}}^+)$. From the predicted anchors, we estimate the transformation $\hat{\xi}$ and correct the image as $x_{\text{g}}^{\text{corr}} = \mathcal{W}(x_{\text{g}}^+;\hat{\xi})$. Watermark recovery is then evaluated by inverting and decoding $x_{\text{g}}^{\text{corr}}$. To maintain accurate anchor prediction during residual optimization, we additionally apply

$$\mathcal{L}_{\text{ali}} = \|P_p - P_{\text{gt}}\|_2^2, \tag{5}$$

where P_{gt} is obtained by applying the sampled transformation ξ to the reference anchors. With A fixed, this loss guides the residual to support accurate geometric correction.

Unified Optimization. To simplify notation, we denote the image fed into inversion as

$$x_T(r) = \begin{cases} x_{\text{ng}}^+, & T = T_{\text{ng}}, \\ x_{\text{g}}^{\text{corr}}, & T = T_{\text{g}}. \end{cases} \tag{6}$$

We optimize r using the unified objective:

$$\min_r \ \mathbb{E}_{T\sim\mathcal{P}}\Big[\mathcal{L}_{\text{wm}}\big(D(I(x_T(r);\emptyset)), m\big) + 1[T = T_{\text{g}}]\lambda_{\text{ali}}\mathcal{L}_{\text{ali}}\Big] + \lambda_{\text{img}}\mathcal{L}_{\text{img}}(x^+, x), \tag{7}$$

where $1[T = T_{\text{g}}]$ equals one for geometric distortions and zero otherwise. The weights λ_{ali} and λ_{img} control alignment supervision and residual imperceptibility, respectively. We optimize the residual using Adam for K iterations and release the final compensated image $x^+ = x + r^*$.

Table 1: Overall robustness and visual quality comparison. Robustness results are averaged within each distortion family and reported as SD-v1.4/SD-v2.1.

Methods	Clean		Non-geometric		Geometric		FID	CLIP-Score
	TPR	Bit Acc.	TPR	Bit Acc.	TPR	Bit Acc.		
Stable Diffusion	-	-	-	-	-	-	25.23±0.18	0.3629±0.0006
DwtDct	0.770/0.710	0.818/0.797	0.015/0.015	0.503/0.508	0.000/0.000	0.482/0.484	24.96±0.15	0.3414±0.0632/0.3587±0.0416
DwtDctSVD	**1.000/1.000**	0.999/0.998	0.113/0.120	0.540/0.540	0.000/0.000	0.473/0.472	24.45±0.21	0.3380±0.0621/0.3552±0.0423
RivaGAN	0.950/0.970	0.985/0.990	0.225/0.239	0.722/0.726	0.505/0.513	0.876/0.878	24.25±0.16	0.3390±0.0731/0.3573±0.0430
HiDDeN	**1.000/1.000**	0.980/0.992	0.190/0.219	0.652/0.673	0.011/0.031	0.549/0.584	24.87±0.28	0.3518±0.0015/0.3605±0.0012
MBRS	**1.000/1.000**	**1.000/1.000**	0.286/0.297	0.729/0.744	0.804/0.815	0.843/0.851	24.98±0.26	0.3601±0.0011/0.3605±0.0012
CIN	**1.000/1.000**	**1.000/1.000**	0.466/0.485	0.763/0.784	0.000/0.020	0.524/0.555	25.05±0.24	0.3562±0.0013/0.3608±0.0010
TrustMark	**1.000/1.000**	**1.000/1.000**	0.772/0.800	0.883/0.899	0.091/0.104	0.537/0.576	25.12±0.25	0.3506±0.0011/0.3623±0.0008
Stable Signature	0.941/**1.000**	0.961/0.993	0.223/0.209	0.650/0.650	0.761/0.758	0.857/0.859	25.46±0.17	0.3411±0.0017/0.3623±0.0022
LaWa	**1.000/1.000**	**1.000/1.000**	0.388/0.394	0.813/0.816	0.582/0.575	0.859/0.856	25.43±0.16	0.3504±0.0452/0.3662±0.0402
Tree-Ring	**1.000/1.000**	-	0.863/0.892	-	0.527/0.578	-	25.42±0.12	0.3632±0.0005/0.3651±0.0003
RingID	**1.000/1.000**	-	0.895/0.894	-	0.200/0.204	-	25.42±0.18	0.3523±0.0024/0.3666±0.0011
GS	**1.000/1.000**	**1.000/1.000**	0.953/0.958	0.921/0.920	0.000/0.000	0.502/0.502	25.21±0.20	0.3597±0.0009/0.3637±0.0004
PRC	**1.000/1.000**	**1.000/1.000**	0.256/0.260	0.508/0.503	0.000/0.000	0.522/0.524	25.87±0.18	0.3168±0.0012/0.3218±0.0009
Tree-Ring-PreFix	**1.000/1.000**	-	**0.963/0.980**	-	**0.941/0.951**	-	25.41±0.17	0.3628±0.0007/0.3644±0.0006
RingID-PreFix	**1.000/1.000**	-	**0.936/0.945**	-	**0.968/0.978**	-	25.40±0.19	0.3602±0.0012/0.3654±0.0010
GS-PreFix	**1.000/1.000**	**1.000/1.000**	**0.997/0.998**	**0.958/0.968**	**0.982/0.992**	**0.939/0.944**	25.20±0.22	0.3597±0.0008/0.3637±0.0005
PRC-PreFix	**1.000/1.000**	**1.000/1.000**	**0.926/0.930**	**0.734/0.741**	**1.000/1.000**	**0.919/0.926**	25.79±0.20	0.3170±0.0010/0.3214±0.0010

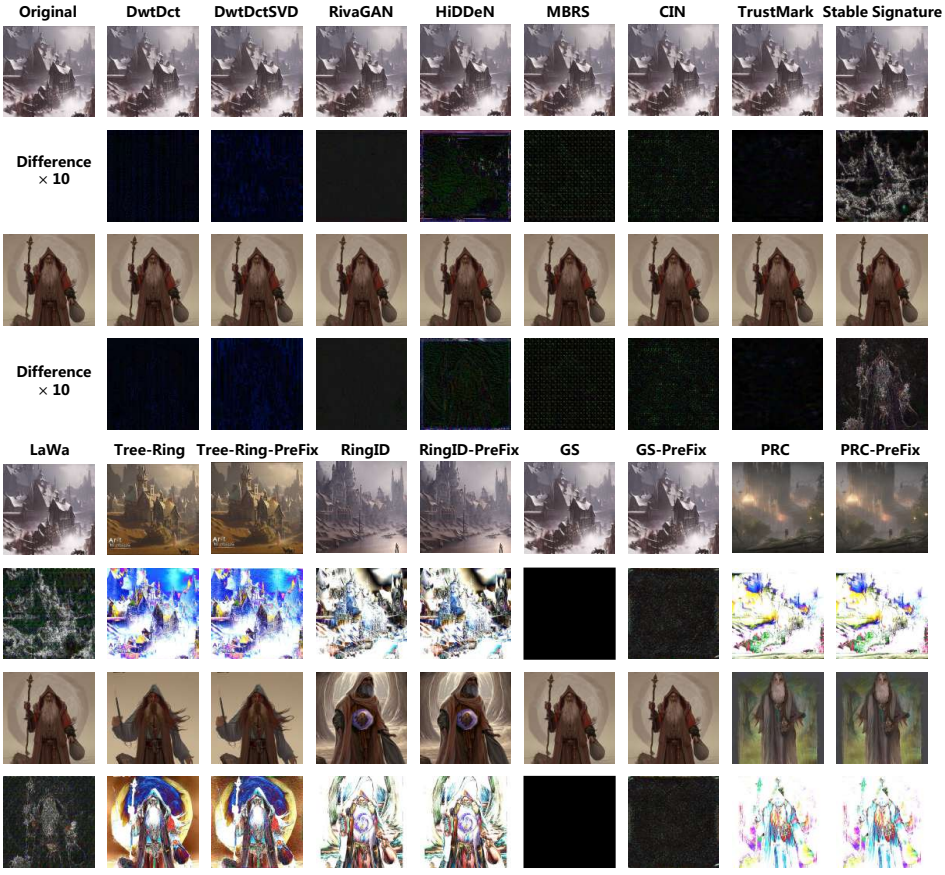


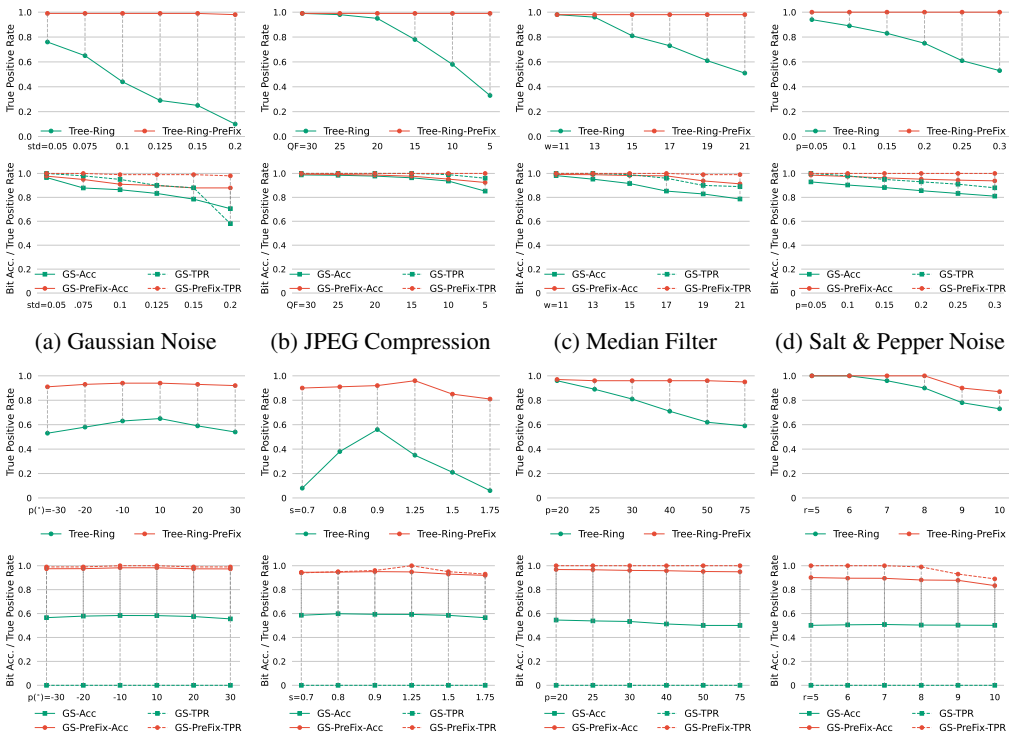
Figure 3: Visual comparison of watermarked and non-watermarked images.

4 Experiments

4.1 Experimental Setup

Implementation details.

We conduct our main experiments on Stable Diffusion (SD) Rombach et al. (2022), specifically SD-v1.4 and SD-v2.1. Images have a resolution of 512×512, with corresponding latent representations of size $4 \times 64 \times 64$. Prompts are sampled from the SD-Prompt dataset Gustavosta (2024). We additionally evaluate PreFix on SDXL-base-1.0 and SD-v3-medium. To train the alignment predictor A, we



(a) Gaussian Noise (b) JPEG Compression (c) Median Filter (d) Salt & Pepper Noise

(e) Rotation (f) Scaling (g) Translation (h) Shear mapping

Figure 4: Robustness of Tree-Ring and GS with and without PreFix under varying distortion strengths.

use COCO Lin et al. (2014) and apply geometric distortions, including rotation ($\theta \in [-45°, 45°]$), translation ($t_x, t_y \in [10, 75]$ pixels), scaling ($s \in [0.5, 2.0]$, with padding or cropping), and shear mapping (5–10 pixels). During residual optimization, we sample distortions from a unified pool $T = T_{\text{ng}} \cup T_{\text{g}}$, where T_{ng} includes Gaussian noise ($\mu = 0, \sigma = 0.15$), median filtering ($w = 15$), and JPEG compression (QF $= 25$), while T_{g} follows the geometric settings used to train A. For SD-v1.4 and SD-v2.1, we perform 50-step DDIM inversion under the $\emptyset$-condition. The residual is optimized for $K = 6$ iterations using Adam, with a learning rate of 2×10^{-2} for the first four iterations and 8×10^{-3} for the remaining two. Additional implementation details are provided in the Appendix.

Baselines. We compare with representative watermarking methods from three categories: (1) post-hoc methods, including DwtDct Cox et al. (2008), DwtDctSVD Cox et al. (2008), RivaGAN Zhang et al. (2019), HiDDeN Zhu et al. (2018), MBRS Jia et al. (2021), CIN Ma et al. (2022), and TrustMark Bui et al. (2025); (2) fine-tuning-based methods, including Stable Signature Fernandez et al. (2023) and LaWa Rezaei et al. (2024); and (3) inversion-based methods, including Tree-Ring Wen et al. (2023), Gaussian Shading (GS) Yang et al. (2024), RingID Ci et al. (2024), and PRC Gunn et al. (2025). We integrate PreFix into all inversion-based methods and compare each method with and without PreFix.

Evaluation metrics. Following Gaussian Shading Yang et al. (2024), we evaluate robustness on 1,000 watermarked images using the true positive rate (TPR) at a theoretical target false positive rate (FPR) of 10^{-6} and extraction bit accuracy (Bit Acc.). To assess generation quality, we report Fréchet Inception Distance (FID) Heusel et al. (2017) and CLIP score Radford et al. (2021), comparing them with those of the corresponding unwatermarked generation baseline.

4.2 Comparison to Baselines

We compare PreFix with baselines in terms of robustness and visual quality. Robustness is evaluated under eight non-geometric degradations: JPEG compression (QF=15), Gaussian noise ($\sigma = 0.05$), median filtering ($w = 15$), salt-and-pepper noise ($p = 0.15$), Gaussian blur (radius $r = 8$), dropout (30%), cropout (70%), and brightness (factor=6); and five geometric transformations: rotation (30°), translation (50 pixels), scaling (0.75× with padding and 1.25× with cropping), and shear mapping (5 pixels). We apply PreFix to Tree-Ring, RingID, GS, and PRC. Table 1 reports their performance with

Table 2: Evaluation of GS on SDXL-base-1.0 and SD-v3-medium.

Methods	Clean		Non-geometric		Geometric		FID ↓	CLIP Score ↑
	TPR ↑	Bit Acc. ↑	TPR ↑	Bit Acc. ↑	TPR ↑	Bit Acc. ↑		
SDXL-base-1.0	-	-	-	-	-	-	25.36±0.16	0.3668±0.0004
GS	1.000	0.912	0.920	0.861	0.000	0.508	25.27±0.19	0.3642±0.0004
GS-PreFix	1.000	**0.953**	**0.985**	**0.916**	**0.976**	**0.967**	25.23±0.23	0.3639±0.0006
SD-v3-medium	-	-	-	-	-	-	25.30±0.12	0.3672±0.0002
GS	1.000	0.995	0.931	0.885	0.000	0.501	25.26±0.15	0.3651±0.0003
GS-PreFix	1.000	**0.998**	**0.986**	**0.929**	**0.972**	**0.964**	25.25±0.18	0.3650±0.0005

and without PreFix alongside other baselines, with robustness results averaged within each distortion family. Per-distortion results and distortion examples are provided in the Appendix.

Across both SD-v1.4 and SD-v2.1, PreFix improves the average detection TPR of all four inversion-based methods under both distortion families, as well as the bit accuracy of GS and PRC. The gains are particularly pronounced under geometric transformations: averaged across the two SD versions, PreFix improves the bit accuracy of GS and PRC by 43.95 and 39.95 percentage points, respectively, while also substantially increasing the detection TPR of Tree-Ring and RingID. These results demonstrate the effectiveness of PreFix across different inversion-based watermarking mechanisms.

PreFix maintains FID and CLIP scores comparable to the corresponding base methods, indicating that the robustness gains largely preserve generation quality and image–text alignment. Fig. 3 further shows that compensated images remain visually similar to their counterparts before applying PreFix.

We further evaluate Tree-Ring and GS under varying distortion strengths. For non-geometric degradations, we vary Gaussian noise ($\sigma \in [0.05, 0.2]$), JPEG compression (QF from 5 to 30), median filtering (w from 11 to 21), and salt-and-pepper noise ($p \in [0.05, 0.3]$). For geometric transformations, we vary rotation (-30° to 30°), scaling ($0.7\times$ to $1.75\times$), translation (20–75 pixels), and shear mapping (5–10 pixels). Fig. 4 shows that PreFix generally improves detection TPR for both methods and bit accuracy for GS across the tested distortion strengths.

Table 3: Bit accuracy of GS-PreFix under different generation conditions.

Settings	Guidance Scale			Sampling step		
	3	7.5	11	10	25	50
w/o Distortion	1.000	1.000	1.000	1.000	1.000	1.000
Non-geo Distortion	0.958	0.968	0.946	0.964	0.962	0.968
Geo Distortion	0.949	0.944	0.931	0.941	0.944	0.944

Settings	Sampling methods					Average
	DDIM	UniPC	PNDM	DEIS	DPMSolver	
w/o Distortion	1.000	1.000	1.000	1.000	1.000	1.000
Non-Geo Distortion	0.961	0.960	0.960	0.963	0.968	0.962
Geo Distortion	0.941	0.938	0.942	0.939	0.944	0.941

4.3 Generalization across Generative Models and Conditions

Different Generative Models. We evaluate PreFix on SDXL-base-1.0 and SD-v3-medium using GS. We use DDIM inversion for SDXL and Euler-based ODE inversion for SD-v3-medium. Per-distortion results are provided in the Appendix. As shown in Table 2, PreFix improves detection TPR and bit accuracy under both distortion types on both models. Under geometric transformations, bit accuracy increases by 45.90 and 46.30 percentage points on SDXL and SD-v3-medium, respectively. These results demonstrate the generalization of PreFix across generative architectures.

Different Generation Conditions. We evaluate GS-PreFix under different guidance scales, sampling steps, and samplers. Specifically, we test guidance scales of 3, 7.5, and 11; sampling steps of 10, 25, and 50; and five samplers: DDIM Song et al. (2020), UniPC Zhao et al. (2023a), PNDM Liu et al. (2022), DEIS Zhang & Chen (2022), and DPMSolver Lu et al. (2022). The default configuration uses a guidance scale of 7.5, 50 sampling steps, and DPMSolver. We vary one factor at a time while keeping the remaining settings fixed. As shown in Table 3, GS-PreFix maintains perfect bit accuracy on clean images and high bit accuracy under both distortion families across the tested conditions.

Table 4: Robustness of GS-PreFix under reconstruction and purification attacks.

Metric	Reconstruction Attack				
	Cheng	Bmshj	VQ-VAE	KL-VAE	SD-v1.4
PSNR (dB)	34.29	35.32	29.78	29.90	26.41
TPR	0.970	0.910	0.970	0.980	0.970
Bit Acc.	0.935	0.873	0.929	0.940	0.915
Metric	Purification Attack				
	$s = 0.05$	$s = 0.1$	$s = 0.3$	$s = 0.5$	$s = 0.7$
PSNR (dB)	25.93	24.08	20.72	19.10	15.92
TPR	0.980	0.980	0.940	0.520	0.000
Bit Acc.	0.944	0.912	0.873	0.743	0.648

Table 5: Robustness of GS-PreFix under combined distortions.

Metric	R-15° S-1.25x	R-15° S-0.9x	R-15° T-25	S-1.25x T-25	S-0.9x T-25
TPR	0.990	1.000	0.970	0.960	0.960
Bit Acc.	0.937	0.946	0.918	0.902	0.905
Metric	R-15° JPEG	R-15° GauN	R-15° MedF	T-25 JPEG	S-1.25x JPEG
TPR	0.990	1.000	1.000	0.990	0.980
Bit Acc.	0.943	0.950	0.955	0.931	0.925

4.4 Robustness under Combined Distortions

We further evaluate GS-PreFix under 10 combined distortion settings, including combinations of geometric transformations and combinations of geometric and non-geometric distortions. As summarized in Table 5, "R", "S", and "T" denote rotation, scaling, and translation, respectively, while "JPEG", "GauN", and "MedF" denote JPEG compression (QF=75), Gaussian noise ($\sigma = 0.05$), and median filtering ($w = 7$). GS-PreFix achieves an average detection TPR of 98.40% and an average bit accuracy of 93.12% across these settings, demonstrating its robustness under combined distortions.

4.5 Robustness to Adaptive Attacks

We further evaluate PreFix under two adaptive attacks that explicitly attempt to remove watermark information. **(1) Reconstruction attack** Zhao et al. (2023b) reconstructs a watermarked image using an autoencoder or a diffusion model. **(2) Purification attack** Nie et al. (2022) adds random noise to the watermarked image and then applies diffusion-based denoising. For reconstruction attacks, we adopt four widely used autoencoders, including Cheng Cheng et al. (2020), Bmshj Ballé et al. (2018), VQ-VAE, and KL-VAE Rombach et al. (2022). For purification attacks, we add noise with strengths ranging from 0.05 to 0.7 and perform 100-step diffusion denoising to obtain purified images. As shown in Table 4, GS-PreFix achieves at least 91.0% TPR and 87.3% bit accuracy under reconstruction attacks and maintains reliable recovery under mild purification ($s \leq 0.3$). Stronger purification reduces recovery accuracy and degrades image quality, revealing a limitation under aggressive purification. Qualitative results are provided in the Appendix.

Table 6: Ablation of degradation-pool optimization (DO) and alignment-aware optimization (AO) on SD-v2.1. Results are reported as TPR for Tree-Ring and RingID, and TPR/Bit Acc. for GS and PRC.

Methods	Mechanisms	Non-geometric					Geometric				
		JPEG	GauNoise	MedFilter	SPNoise	GauBlur	Rotation	Scale 0.75x	Scale 1.25x	Translation	Shear
Tree-Ring	*Base*	0.844	0.760	0.881	0.790	1.000	0.547	0.097	0.356	0.892	1.000
	DO	0.992	1.000	0.986	1.000	1.000	0.537	0.081	0.365	0.904	1.000
	AO	0.980	0.743	0.878	0.776	1.000	0.931	0.926	0.069	0.972	1.000
	PreFix	0.992	1.000	0.988	1.000	1.000	0.924	0.903	0.963	0.964	1.000
RingID	*Base*	0.932	0.883	0.826	0.879	0.802	1.000	0.020	0.000	0.000	0.000
	DO	0.982	0.946	0.891	0.941	0.871	0.997	0.026	0.004	0.006	0.003
	AO	0.935	0.879	0.832	0.873	0.809	0.999	0.981	0.978	0.984	0.975
	PreFix	0.986	0.942	0.888	0.938	0.864	0.998	0.975	0.972	0.979	0.968
GS	*Base*	0.948/0.916	0.943/0.878	0.902/0.907	0.928/0.880	0.956/0.813	0.000/0.503	0.000/0.505	0.000/0.501	0.000/0.501	0.000/0.502
	DO	1.000/0.984	1.000/0.987	1.000/0.987	1.000/0.969	1.000/0.923	0.000/0.537	0.000/0.573	0.000/0.582	0.000/0.566	0.000/0.589
	AO	0.963/0.932	0.955/0.869	0.990/0.903	0.977/0.872	0.951/0.806	1.000/0.991	0.960/0.952	1.000/0.944	1.000/0.956	1.000/0.919
	PreFix	1.000/0.977	1.000/0.977	1.000/0.980	1.000/0.956	1.000/0.891	0.995/0.985	0.965/0.945	1.000/0.938	1.000/0.952	1.000/0.901
PRC	*Base*	0.448/0.502	0.054/0.499	0.748/0.500	0.072/0.499	0.070/0.499	0.000/0.551	0.000/0.507	0.000/0.523	0.000/0.542	0.000/0.498
	DO	0.947/0.751	0.915/0.714	0.963/0.786	0.919/0.706	0.895/0.697	0.000/0.558	0.000/0.529	0.000/0.547	0.000/0.561	0.000/0.526
	AO	0.456/0.508	0.061/0.504	0.739/0.497	0.078/0.503	0.074/0.502	1.000/0.971	1.000/0.941	1.000/0.923	1.000/0.937	1.000/0.909
	PreFix	0.941/0.743	0.902/0.702	0.958/0.777	0.911/0.694	0.886/0.684	1.000/0.962	1.000/0.932	1.000/0.912	1.000/0.926	1.000/0.897

4.6 Ablation and Analysis

Effectiveness of DO and AO. We evaluate the contributions of degradation-pool optimization (DO) and alignment-aware optimization (AO) on SD-v2.1 using Tree-Ring, RingID, GS, and PRC. For each method, we compare four settings: (1) Base, without either mechanism; (2) DO, with degradation-pool optimization only; (3) AO, with alignment-aware optimization only; and (4) PreFix, combining both. All settings use DDIM inversion under the $\emptyset$-condition and the original extraction mechanism of the corresponding watermarking method. As shown in Table 6, DO consistently improves



Figure 5: Visual examples of geometric correction under different transformations.

non-geometric robustness, while AO mainly improves geometric robustness through pre-inversion alignment. Although individual mechanisms can outperform PreFix under specific distortions, PreFix provides more consistent robustness across both distortion families, supporting the complementary roles of DO and AO.

Geometric Correction. Fig. 5 visualizes distorted images and their corrected counterparts under different geometric transformations. Correction based on predicted anchors produces results visually consistent with those obtained using ground-truth transformations, illustrating how pre-inversion alignment compensates for spatial misalignment.

Table 7: Generation time and average bit accuracy under different optimization budgets on SD-v2.1.

Metric	GS	GS-PreFix					
		$K=0$	$K=1$	$K=2$	$K=4$	$K=6$	$K=8$
Time (s)	7.12	7.29	10.75	14.25	21.25	28.14	35.10
Non-geo.	0.920	0.919	0.925	0.935	0.950	0.968	0.969
Geo.	0.502	0.923	0.929	0.936	0.942	0.944	0.944

Computational Cost. We evaluate PreFix on SD-v2.1 with GS using an NVIDIA H100 GPU. Table 7 reports the total generation time per image and average bit accuracy under different optimization budgets. At $K = 0$, PreFix retains template initialization and pre-inversion alignment without residual optimization. Generation time increases only from 7.12s to 7.29s, while geometric bit accuracy improves from 0.502 to 0.923 and non-geometric accuracy remains nearly unchanged. At $K = 6$, generation takes 28.14s, approximately $3.95\times$ that of GS, with non-geometric and geometric bit accuracy reaching 0.968 and 0.944, respectively. These correspond to gains of 4.8 and 44.2 percentage points over GS. Increasing the budget to $K = 8$ requires another 6.96s but improves non-geometric accuracy by only 0.1 percentage points, with no change in geometric accuracy at the reported precision. We therefore use $K = 6$ as the default budget to balance robustness and computational cost, while smaller budgets offer lower-cost alternatives. Per-distortion results are provided in the Appendix.

5 Conclusion

In this paper, we propose PreFix, a generation-side pre-compensation framework for robust inversion-based image watermarking. PreFix optimizes an imperceptible residual for each watermarked image before release, improving extraction robustness without modifying the underlying inversion algorithm. Its unified residual optimization combines degradation-pool optimization for reliable recovery under non-geometric degradations with alignment-aware optimization to support transformation estimation and pre-inversion correction under geometric distortions. Extensive experiments across watermarking methods and generative models demonstrate consistent robustness improvements under both non-geometric and geometric distortions.

AI USE STATEMENT

We used ChatGPT to assist with drafting and polishing the manuscript, improving its organization, formatting LaTeX tables, and editing code. We also used it to obtain feedback on experimental descriptions and discuss the interpretation and presentation of results. All AI-assisted text, suggestions, and code were reviewed by the authors, and all reported experimental results were obtained from our experiments. The authors take full responsibility for the final content of this paper, including its methodology, results, and conclusions.

REFERENCES

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

Tu Bui, Shruti Agarwal, and John Collomosse. Trustmark: Robust watermarking and watermark removal for arbitrary resolution images. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18629–18639. IEEE, 2025.

Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7939–7948, 2020.

Hai Ci, Pei Yang, Yiren Song, and Mike Zheng Shou. Ringid: Rethinking tree-ring watermarking for enhanced multi-key identification. In *European Conference on Computer Vision*, pp. 338–354. Springer, 2024.

Ingemar J Cox, Matthew L Miller, Jeffrey A Bloom, Jessica Fridrich, and Ton Kalker. Digital watermarking. *Morgan Kaufmann Publishers*, 54:56–59, 2008.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

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.

Samuel Gunn, Xuandong Zhao, and Dawn Song. An undetectable watermark for generative image models. In *International Conference on Learning Representations*, volume 2025, pp. 6612–6637, 2025.

Gustavosta. Stable-diffusion-prompts dataset. Hugging Face, 2024. URL `https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts`. Accessed: 12-Dec-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. *Advances in neural information processing systems*, 30, 2017.

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

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*, pp. 41–49, 2021.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6007–6017, 2023.

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. Springer, 2014.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.

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, 2022.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Ahmad Rezaei, Mohammad Akbari, Saeed Ranjbar Alvar, Arezou Fatemi, and Yong Zhang. Lawa: Using latent space for in-generation image watermarking. In *European Conference on Computer Vision*, pp. 118–136. Springer, 2024.

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*, pp. 10684–10695, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

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.

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.

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

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:49842–49869, 2023a.

Xuandong Zhao, Kexun Zhang, Yu-Xiang Wang, and Lei Li. Generative autoencoders as watermark attackers: Analyses of vulnerabilities and threats. 2023b.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *European conference on computer vision*, pp. 682–697. Springer, 2018.

A THE IMPLEMENTATION DETAILS OF COMPARED BASELINE METHODS

We compare PreFix with several state-of-the-art generative image watermarking methods, categorized into three groups. For all methods, we utilize their officially released open-source implementations and configure the watermark parameters based on their respective specifications.

(1) Image watermarking-based methods, including DwtDct Cox et al. (2008), DwtDctSVD Cox et al. (2008), RivaGAN Zhang et al. (2019), HiDDeN Zhu et al. (2018), MBRS Jia et al. (2021), CIN Ma et al. (2022), and TrustMark Bui et al. (2025). For DwtDct, DwtDctSVD, and RivaGAN, we use the implementations provided in the invisible-watermark repository[1], with watermark capacities of 256, 256, and 32 bits, respectively. For HiDDeN[2], MBRS[3], and CIN[4], we use their publicly available implementations with watermark capacities of 30, 256, and 30 bits, respectively. For TrustMark[5], we use the official implementation with a raw watermark length of 100 bits.

(2) Fine-tuning-based methods, including Stable Signature Fernandez et al. (2023) and LaWa Rezaei et al. (2024). We adopt the open-source code of Stable Signature[6], setting the watermark capacity to 48 bits. Similarly, we use the publicly available code for LaWa[7], also setting the capacity to 48 bits.

(3) Inversion-based methods, Tree-Ring Wen et al. (2023), RingID Ci et al. (2024), Gaussian Shading (GS) Yang et al. (2024), and PRC Gunn et al. (2025). For Tree-Ring[8], we set `w_channel`=0, `w_pattern`=`ring`, and `w_radius`=10, and evaluate watermark detection. For RingID[9], we set `ring_width`=1, `quantization_levels`=2, and `ring_value_range`=64. For GS[10], we set $f_c = 1$, $f_{hw} = 8$, and $l = 1$, yielding a watermark capacity of 256 bits. For PRC[11], we use the publicly available implementation with the default message length of 256 bits.

B TPR CALCULATION WITH MULTI-BITS METHODS

To calculate the TPR of multi-bits methods, we adopt the same TPR calculation method proposed by GS(Yang et al., 2024), which can be described as follows: Assume the watermark to be embedded is $w \in \{0,1\}^k$ where k is the length. For the image to be detected x, we can extract the watermark w_x from the x and calculate the bit accuracy of w_x, noted as $Acc(w_x, w)$. Then a threshold τ is applied to make a decision: if

$$Acc(w_x, w) \geq \tau,$$

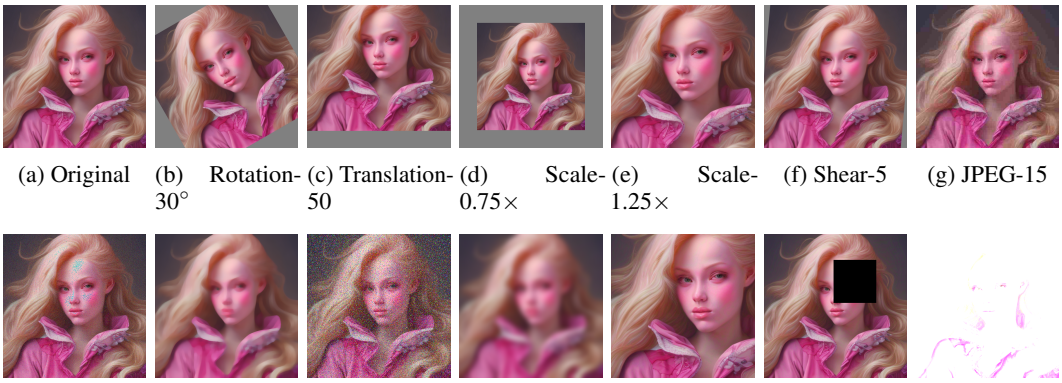
x is regarded as watermarked. In this form, τ is set based on a required estimated false positive rate (FPR), which is defined as the probability that $Acc(w_{x'}, w)$ of a non-watermarked image x' exceeds the threshold τ. Such a probability can be further calculated with regularized incomplete beta function $B_x(a; b)$Fernandez et al. (2023):

$$\begin{aligned}\mathrm{FPR}(\tau) &= \mathcal{P}\left(Acc\left(w_{x'}, w\right) > \tau\right) = \frac{1}{2^k}\sum_{i=\tau+1}^{k}\binom{k}{i} \\ &= B_{1/2}(\tau+1, k-\tau).\end{aligned}$$

Detail analysis of the TPR calculation can be found in (Yang et al., 2024).

Shortly, to calculate the TPR, we first set an FPR (e.g. 10^{-6} in this paper). Then according to the set FPR, we can determine a threshold τ with the embedded watermark length k. The image with extraction bit accuracy larger than τ is regarded as watermarked.

[1] https://github.com/ShieldMnt/invisible-watermark
[2] https://github.com/ando-khachatryan/HiDDeN
[3] https://github.com/jzyustc/MBRS
[4] https://github.com/rmpku/CIN
[5] https://github.com/adobe/trustmark
[6] https://github.com/facebookresearch/stable_signature
[7] LaWa Official Code
[8] https://github.com/YuxinWenRick/tree-ring-watermark
[9] https://github.com/showlab/RingID
[10] https://github.com/bsmhmmlf/Gaussian-Shading
[11] https://github.com/xuandongzhao/prc-watermark



(a) Original (b) Rotation-30° (c) Translation-50 (d) Scale-0.75× (e) Scale-1.25× (f) Shear-5 (g) JPEG-15

(h) GauNoise-0.05 (i) MedFilter-15 (j) S&P-0.15 (k) GauBlur-8 Cropout-70% Dropout-30% Brightness-6

Figure 6: Visual examples of distortions.

Table 8: Detection TPR of baseline methods under each distortion. Each entry reports results for SD-v1.4/SD-v2.1.

Distortion	DwtDct	DwtDctSVD	RivaGAN	HiDDeN	MBRS	CIN	TrustMark	Stable Signature	LaWa
JPEG	0.000/0.000	0.000/0.000	0.000/0.000	0.000/0.012	0.107/0.122	0.041/0.046	0.624/0.673	0.000/0.000	0.142/0.126
GauBlur	0.000/0.000	0.000/0.030	0.000/0.000	0.027/0.063	0.081/0.090	0.000/0.008	0.823/0.842	0.000/0.000	0.000/0.000
MedFilter	0.000/0.000	0.849/0.857	0.102/0.155	0.032/0.074	0.163/0.172	0.003/0.019	0.757/0.785	0.000/0.000	0.026/0.033
GauNoise	0.000/0.000	0.000/0.000	0.062/0.104	0.054/0.058	0.148/0.156	0.345/0.391	0.884/0.887	0.000/0.000	0.166/0.166
S&P Noise	0.000/0.000	0.000/0.000	0.000/0.010	0.003/0.051	0.197/0.195	0.420/0.425	0.563/0.612	0.000/0.000	0.073/0.076
Brightness	0.121/0.123	0.054/0.069	0.456/0.438	0.467/0.508	0.459/0.474	0.970/0.994	0.765/0.794	0.700/0.610	0.956/0.991
Dropout	0.000/0.000	0.000/0.000	0.182/0.206	0.938/0.986	0.923/0.947	0.964/1.000	0.828/0.861	0.087/0.065	0.743/0.762
Cropout	0.000/0.000	0.000/0.000	1.000/1.000	0.000/0.003	0.208/0.217	0.984/1.000	0.930/0.946	1.000/1.000	1.000/1.000
Average	0.015/0.015	0.113/0.120	0.225/0.239	0.190/0.219	0.286/0.297	0.466/0.485	0.772/0.800	0.223/0.209	0.388/0.394
Rotation	0.000/0.000	0.000/0.000	0.000/0.000	0.000/0.002	0.266/0.283	0.000/0.007	0.000/0.012	0.000/0.000	0.000/0.000
Scale 0.75x	0.000/0.000	0.000/0.000	0.661/0.654	0.025/0.034	0.982/0.988	0.000/0.028	0.001/0.005	0.803/0.806	0.000/0.000
Scale 1.25x	0.000/0.000	0.000/0.000	0.909/0.925	0.023/0.057	0.944/0.958	0.000/0.004	0.005/0.009	1.000/1.000	0.908/0.876
Translation	0.000/0.000	0.000/0.000	0.956/0.987	0.004/0.021	0.920/0.929	0.000/0.037	0.000/0.003	1.000/0.982	1.000/1.000
Shear	0.000/0.000	0.000/0.000	0.000/0.000	0.003/0.043	0.907/0.915	0.000/0.024	0.449/0.491	1.000/1.000	1.000/1.000
Average	0.000/0.000	0.000/0.000	0.505/0.513	0.011/0.031	0.804/0.815	0.000/0.020	0.091/0.104	0.761/0.758	0.582/0.575

C Detailed comparison results

In the overall robustness evaluation, we test watermark reliability under five geometric transformations (rotation $30°$, translation 50 pixels, scaling $0.75\times$ with padding, scaling $1.25\times$ with cropping, and shear mapping 5 pixels) and eight non-geometric degradations (JPEG compression with QF=15, Gaussian noise with $\sigma = 0.05$, median filtering with $w = 15$, salt-and-pepper noise with $p = 0.15$, Gaussian blur with radius $r = 8$, dropout with ratio 30%, cropout with ratio 70%, and brightness adjustment with factor 6). Fig. 6 provides representative visual examples of these distortions.

Detailed results on SD-v1.4 and SD-v2.1. Tables 8, 9, 10 and 11 report per-distortion TPR and bit accuracy, respectively. Each entry is presented as SD-v1.4/SD-v2.1, and averages are computed separately within the non-geometric and geometric distortion families.

The detailed results show that PreFix consistently improves average detection TPR across the evaluated inversion-based methods under both distortion families, while also improving the average bit accuracy of GS and PRC. The gains are particularly pronounced under geometric transformations, highlighting the benefit of alignment before inversion.

Generalization across generative models. We further provide per-distortion results on SDXL-base-1.0 and SD-v3-medium using GS as the base watermarking method. Table 12 reports detection TPR and bit accuracy with and without PreFix, complementing the aggregate results in the main paper. We use DDIM inversion for SDXL and Euler-based ODE inversion for SD-v3-medium to accommodate their respective generative formulations.

Table 9: Detection TPR of inversion-based watermarking methods with and without PreFix under each distortion. Each entry reports results for SD-v1.4/SD-v2.1.

Distortion	Tree-Ring		RingID		GS		PRC	
	Base	+PreFix	Base	+PreFix	Base	+PreFix	Base	+PreFix
JPEG	0.831/0.844	0.949/0.992	0.915/0.932	0.972/0.986	0.936/0.948	1.000/1.000	0.421/0.448	0.939/0.941
GauBlur	0.969/1.000	0.987/1.000	0.802/0.802	0.843/0.864	0.971/0.956	1.000/1.000	0.070/0.070	0.882/0.886
MedFilter	0.860/0.881	0.987/0.988	0.876/0.826	0.883/0.888	0.896/0.902	1.000/1.000	0.744/0.748	0.951/0.958
GauNoise	0.712/0.760	1.000/1.000	0.872/0.883	0.940/0.942	0.937/0.943	1.000/1.000	0.054/0.054	0.898/0.902
S&P Noise	0.732/0.790	1.000/1.000	0.877/0.879	0.930/0.938	0.925/0.928	1.000/1.000	0.069/0.072	0.911/0.911
Brightness	0.857/0.884	0.855/0.883	0.916/0.917	0.965/0.970	0.981/1.000	0.993/1.000	0.560/0.560	0.960/0.963
Dropout	0.971/1.000	0.956/0.994	0.940/0.944	0.981/0.982	0.974/0.988	0.986/0.986	0.052/0.052	0.934/0.934
Cropout	0.972/0.975	0.973/0.986	0.958/0.967	0.977/0.988	1.000/1.000	1.000/1.000	0.076/0.078	0.933/0.946
Average	0.863/0.892	0.963/0.980	0.895/0.894	0.936/0.945	0.953/0.958	0.997/0.998	0.256/0.260	0.926/0.930
Rotation	0.382/0.547	0.922/0.924	1.000/1.000	0.996/0.998	0.000/0.000	0.961/0.995	0.000/0.000	1.000/1.000
Scale 0.75x	0.030/0.097	0.880/0.903	0.000/0.020	0.966/0.975	0.000/0.000	0.947/0.965	0.000/0.000	1.000/1.000
Scale 1.25x	0.342/0.356	0.941/0.963	0.000/0.000	0.968/0.972	0.000/0.000	1.000/1.000	0.000/0.000	1.000/1.000
Translation	0.879/0.892	0.961/0.964	0.000/0.000	0.962/0.979	0.000/0.000	1.000/1.000	0.000/0.000	1.000/1.000
Shear	1.000/1.000	1.000/1.000	0.000/0.000	0.949/0.968	0.000/0.000	1.000/1.000	0.000/0.000	1.000/1.000
Average	0.527/0.578	0.941/0.951	0.200/0.204	0.968/0.978	0.000/0.000	0.982/0.992	0.000/0.000	1.000/1.000

Table 10: Extraction bit accuracy of baseline methods under each distortion. Each entry reports results for SD-v1.4/SD-v2.1.

Distortion	DwtDct	DwtDctSVD	RivaGAN	HiDDeN	MBRS	CIN	TrustMark	Stable Signature	LaWa
JPEG	0.5045/0.5036	0.4561/0.4564	0.5909/0.5809	0.5317/0.5600	0.6461/0.6613	0.6211/0.6380	0.8517/0.8550	0.5690/0.5710	0.7946/0.7910
GauBlur	0.4978/0.5035	0.5577/0.5533	0.5381/0.5422	0.6260/0.6460	0.6088/0.6730	0.4804/0.4930	0.8786/0.9100	0.5690/0.5710	0.5802/0.5827
MedFilter	0.5046/0.5106	0.7633/0.7771	0.7288/0.7522	0.6448/0.6470	0.7054/0.7019	0.5256/0.5350	0.8738/0.8900	0.4619/0.4704	0.7777/0.7763
GauNoise	0.4781/0.4728	0.5253/0.5328	0.7175/0.7466	0.6044/0.6380	0.6944/0.7001	0.8010/0.8060	0.9213/0.9240	0.5352/0.5419	0.7829/0.8010
S&P Noise	0.5005/0.5011	0.5122/0.4996	0.7013/0.7044	0.6096/0.6370	0.7018/0.7163	0.7839/0.8060	0.8178/0.8400	0.5365/0.5346	0.7477/0.7548
Brightness	0.5100/0.5150	0.4970/0.4890	0.8406/0.8237	0.8253/0.8290	0.8016/0.8093	0.9458/0.9920	0.8848/0.8900	0.9017/0.8835	0.9871/0.9929
Dropout	0.5160/0.5423	0.5078/0.5019	0.6722/0.6779	0.9547/0.9880	0.9604/0.9672	0.9962/1.0000	0.8777/0.9200	0.6472/0.6446	0.8301/0.8317
Cropout	0.5142/0.5162	0.4995/0.5074	0.9873/0.9774	0.4174/0.4420	0.7172/0.7229	0.9504/1.0000	0.9576/0.9610	0.9791/0.9802	1.0000/1.0000
Average	0.5032/0.5081	0.5399/0.5397	0.7221/0.7257	0.6517/0.6734	0.7295/0.7440	0.7631/0.7838	0.8829/0.8988	0.6500/0.6497	0.8125/0.8163
Rotation	0.4996/0.5071	0.4934/0.4877	0.6303/0.6278	0.4427/0.4920	0.7028/0.7350	0.4825/0.5160	0.5003/0.5350	0.5603/0.5609	0.7323/0.7281
Scale 0.75x	0.4755/0.4762	0.4900/0.4912	0.9025/0.9019	0.5741/0.6160	0.9011/0.9020	0.5443/0.5930	0.4788/0.5210	0.8691/0.8655	0.6338/0.6302
Scale 1.25x	0.4864/0.4879	0.4513/0.4461	0.9697/0.9725	0.6110/0.6470	0.8990/0.9020	0.5040/0.5100	0.4853/0.5150	0.9310/0.9407	0.9410/0.9388
Translation	0.4623/0.4624	0.4427/0.4477	0.9794/0.9856	0.5514/0.5760	0.8689/0.8706	0.5700/0.5910	0.4614/0.5070	0.9332/0.9381	0.9921/0.9902
Shear	0.4843/0.4882	0.4883/0.4862	0.9004/0.9037	0.5647/0.5890	0.8424/0.8440	0.5195/0.5640	0.7602/0.8030	0.9912/0.9919	0.9956/0.9919
Average	0.4816/0.4844	0.4731/0.4718	0.8765/0.8783	0.5488/0.5840	0.8428/0.8507	0.5241/0.5548	0.5372/0.5762	0.8570/0.8594	0.8590/0.8558

D DETAILS OF ADAPTIVE ATTACKS EXPERIMENTS

Section 4.5 evaluates PreFix under adaptive attacks that explicitly aim to remove watermark information. In this context, we consider an attack to be successful only if it satisfies two requirements: (i) visual consistency, meaning the attacked image remains visually close to the original watermarked image; and (ii) watermark removal, meaning the watermark becomes undetectable from the attacked image.

We consider two representative adaptive attacks: reconstruction attacks and purification attacks. As reported in Table 4 of the main paper, PreFix maintains high robustness against reconstruction attacks. We additionally analyze the purification attack, where an adversary first injects random noise into the watermarked image and then applies diffusion-based denoising to remove both the added noise and the watermark. PreFix remains robust under mild purification strengths (e.g., $s \in [0.05, 0.3]$), while stronger purification leads to a noticeable performance drop.

Fig.7 provides qualitative results under purification. When the purification strength is small (e.g., s=0.05), the attacked image preserves high visual similarity to the original. As s increases, the denoising process increasingly alters image appearance; at s=0.7, both fine details and global structures are substantially changed. Although such aggressive purification can further suppress the watermark signal, it no longer satisfies the visual-consistency requirement and therefore does not constitute a practical "successful" attack under our evaluation criteria. Consistent with Table 4, even

Table 11: Extraction bit accuracy of GS and PRC with and without PreFix under each distortion. Each entry reports results for SD-v1.4/SD-v2.1.

Distortion	GS		PRC	
	Base	+PreFix	Base	+PreFix
JPEG	0.9103/0.9155	0.9537/0.9771	0.4984/0.5017	0.7319/0.7426
GauBlur	0.8472/0.8132	0.9017/0.8905	0.5038/0.4991	0.6790/0.6843
MedFilter	0.9221/0.9072	0.9773/0.9797	0.5173/0.4997	0.7638/0.7768
GauNoise	0.8571/0.8776	0.9396/0.9774	0.5239/0.4985	0.6984/0.7015
S&P Noise	0.8752/0.8798	0.9332/0.9557	0.5053/0.4992	0.6905/0.6937
Brightness	0.9722/0.9815	0.9715/0.9809	0.5185/0.5259	0.8006/0.8014
Dropout	0.9872/0.9896	0.9891/0.9895	0.4988/0.5004	0.7472/0.7589
Cropout	0.9934/0.9951	0.9962/0.9946	0.5007/0.4985	0.7573/0.7698
Average	0.9206/0.9199	0.9578/0.9682	0.5083/0.5029	0.7336/0.7411
Rotation	0.4998/0.5031	0.9816/0.9846	0.5349/0.5510	0.9608/0.9616
Scale 0.75x	0.5079/0.5047	0.9409/0.9451	0.5137/0.5070	0.9288/0.9324
Scale 1.25x	0.5051/0.5014	0.9333/0.9382	0.5175/0.5230	0.9015/0.9117
Translation	0.4984/0.5007	0.9557/0.9516	0.5382/0.5420	0.9160/0.9258
Shear	0.4974/0.5015	0.8814/0.9009	0.5072/0.4980	0.8886/0.8969
Average	0.5017/0.5023	0.9386/0.9441	0.5223/0.5242	0.9191/0.9257

Table 12: Per-distortion TPR/Bit Acc. on SDXL-base-1.0 and SD-v3-medium.

Distortion	SDXL-base-1.0		SD-v3-medium	
	GS	GS-PreFix	GS	GS-PreFix
JPEG	0.910/0.852	0.994/0.944	0.930/0.880	0.997/0.955
GauBlur	0.875/0.776	0.974/0.867	0.892/0.804	0.976/0.874
MedFilter	0.921/0.846	0.991/0.936	0.941/0.876	0.994/0.948
GauNoise	0.871/0.765	0.971/0.853	0.882/0.794	0.973/0.864
S&P Noise	0.858/0.760	0.968/0.842	0.871/0.790	0.970/0.858
Brightness	0.963/0.942	0.987/0.938	0.976/0.970	0.995/0.972
Dropout	0.984/0.986	0.998/0.983	0.989/0.994	0.996/0.996
Cropout	0.978/0.957	0.997/0.961	0.967/0.968	0.987/0.968
Average	0.920/0.861	0.985/0.916	0.931/0.885	0.986/0.929
Rotation	0.000/0.512	0.970/0.962	0.000/0.504	0.973/0.963
Scale 0.75x	0.000/0.508	0.987/0.971	0.000/0.501	0.979/0.967
Scale 1.25x	0.000/0.509	0.978/0.968	0.000/0.502	0.968/0.964
Translation	0.000/0.503	0.984/0.974	0.000/0.498	0.981/0.966
Shear	0.000/0.506	0.961/0.958	0.000/0.499	0.959/0.962
Average	0.000/0.508	0.976/0.967	0.000/0.501	0.972/0.964

at a moderate strength s=0.3, the watermark remains extractable, indicating that PreFix provides robustness to purification attacks in the visually consistent regime.

E Details of Alignment Network

As described in Section 3.2.1, we train a lightweight alignment predictor $A(\cdot)$ to estimate geometric anchor locations from a distorted observation, enabling canonicalization prior to DDIM inversion. In our implementation, $A(\cdot)$ is trained with a universal anchor template e shared across images,

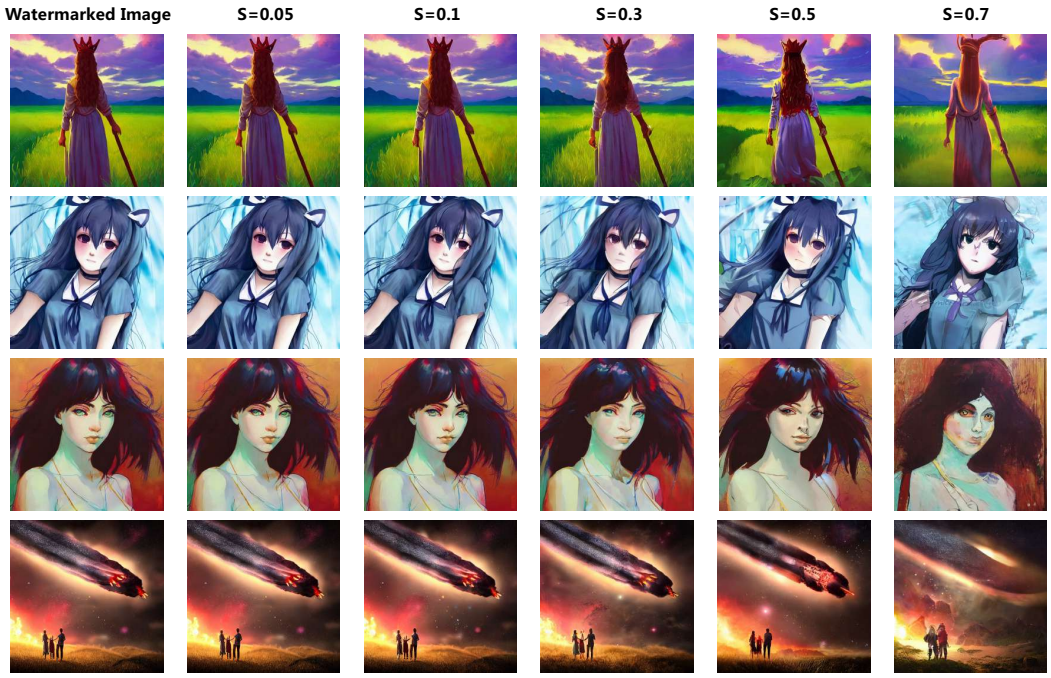


Figure 7: Visual examples of purification attacks at different noise strengths.

which injects weak, alignment-friendly cues to make anchor regression reliable under geometric distortions. In this section, we first describe the network architecture of $A(\cdot)$, and then provide qualitative examples of the predicted anchors and the resulting geometric correction.

The alignment predictor $A(\cdot)$ takes a distorted RGB image as input and outputs the 2D coordinates of a predefined anchor set. Specifically, given an input image $x \in \mathbb{R}^{3\times H\times W}$, $A(\cdot)$ produces four anchor points $\hat{P} \in \mathbb{R}^{4\times 2}$ (i.e., 8 scalars), where each point corresponds to an (x, y) coordinate in the image plane. In our pipeline, these predicted anchors are used to recover the transformation parameters $\hat{\xi}$, and the distorted image is then warped back to a canonical coordinate system for inversion-based watermark extraction.

Architecturally, $A(\cdot)$ is a compact ResNet-style regression network composed of (i) an initial convolutional stem, (ii) two stages of residual blocks with progressive downsampling, and (iii) a shallow regression head:

- **Stem.** A 3×3 convolution followed by BatchNorm and ReLU maps the input from 3 channels to 64 channels (`SingleConv`(3→64, stride 1)).

- **Residual feature extractor.** Two sequential stages (`layer1` and `layer2`) are built from basic residual blocks. Each residual block contains two 3×3 convolutions with BatchNorm and ReLU, plus an identity shortcut; when the spatial resolution changes, a 1×1 convolutional shortcut is used. Across these blocks, the network performs multiple strided convolutions (stride 2) to progressively reduce the spatial resolution while keeping the feature dimension at 64 channels.

- **Regression head.** A final 1×1 convolution projects the 64-channel feature map to a single-channel response map. The response map is flattened into a 256-dimensional vector and fed into a fully connected layer to regress 8 values, which are reshaped into four 2D anchor coordinates (4×2).

F Optimization process of the distortion-aware pre-compensation mechanism

We follow the distortion-aware pre-compensation objective in Section 3.2.2 and optimize a per-image residual r via a few steps of gradient-based refinement. In practice, instead of initializing r from zeros, we warm-start the optimization with the universal anchor template e learned during the training of the alignment predictor $A(\cdot)$ (Section 3.2.1), i.e., $r^{(0)} \leftarrow e$. This initialization provides alignment-friendly cues from the first iteration, improving optimization stability and speeding up convergence, while the subsequent refinement adapts r to the specific image content under the mixed distortion sampling strategy described in the main paper.

We run the optimization for K=6 iterations. Fig. 8 and 9 illustrate the full optimization process on Tree-Ring and GS, respectively, including (1) the original watermarked image, (2) the PreFix output, (3) the optimized residual (amplified by $\times 10$ for visualization), (4) the watermark extraction loss curve, and (5) the alignment loss curve across iterations.

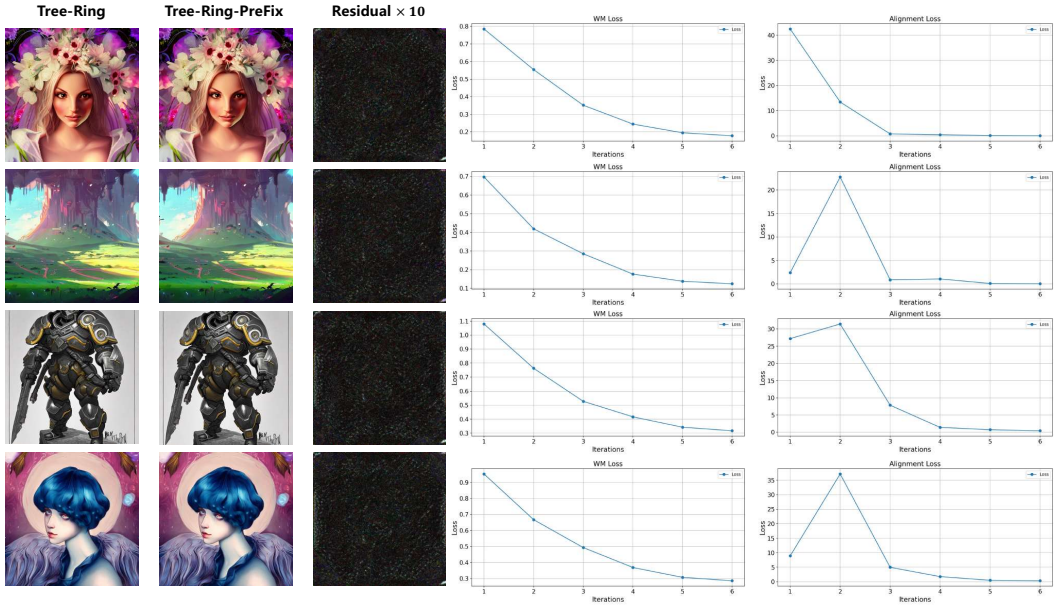


Figure 8: Visualization of the optimization process of Tree-Ring.

As shown in the loss curves, the watermark extraction loss decreases monotonically across iterations, indicating that the residual optimization consistently improves the decoding objective under the sampled distortions. In contrast, the alignment loss can exhibit a transient increase in early iterations before converging. A plausible reason is that r is optimized with a mixed objective: at the beginning, updates driven by $\mathcal{L}_{\text{wm}}$ may reshape the residual toward inversion/decoding-favorable directions that are not yet optimal for geometric regression, temporarily perturbing the anchor cues used by $A(\cdot)$ and leading to a higher alignment error. As optimization proceeds, the alignment loss guides r toward cues that facilitate anchor prediction by the fixed alignment predictor, reducing the alignment error and stabilizing the loss.

G Detailed Results under Different Optimization Budgets

Table 13 provides per-distortion bit accuracy for GS and GS-PreFix on SD-v2.1 under optimization budgets $K \in \{0, 1, 2, 4, 6, 8\}$, complementing the computational cost and average robustness results in Table 7 of the main paper. At $K = 0$, PreFix retains template initialization and pre-inversion alignment without residual optimization.

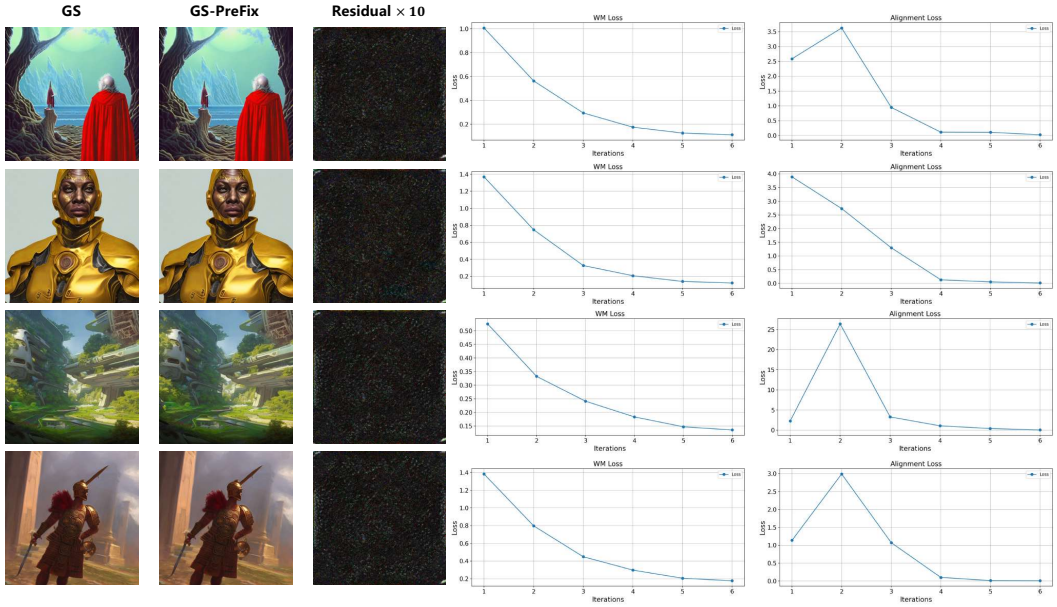


Figure 9: Visualization of the optimization process of GS.

Table 13: Per-distortion bit accuracy under different optimization budgets on SD-v2.1.

Distortion	GS	GS-PreFix					
		$K=0$	$K=1$	$K=2$	$K=4$	$K=6$	$K=8$
JPEG	0.916	0.912	0.924	0.941	0.955	0.977	0.978
GauBlur	0.813	0.812	0.818	0.828	0.860	0.891	0.892
MedFilter	0.907	0.905	0.916	0.931	0.950	0.980	0.980
GauNoise	0.878	0.878	0.893	0.915	0.945	0.977	0.975
S&P Noise	0.880	0.877	0.891	0.910	0.935	0.956	0.956
Brightness	0.982	0.980	0.977	0.975	0.975	0.981	0.981
Dropout	0.990	0.989	0.988	0.988	0.988	0.990	0.990
Cropout	0.995	0.995	0.994	0.994	0.995	0.995	0.996
Average	0.920	0.919	0.925	0.935	0.950	0.968	0.969
Rotation	0.503	0.955	0.966	0.978	0.983	0.985	0.984
Scale 0.75x	0.505	0.900	0.912	0.925	0.942	0.945	0.944
Scale 1.25x	0.501	0.925	0.929	0.933	0.936	0.938	0.938
Translation	0.501	0.942	0.945	0.948	0.950	0.952	0.953
Shear	0.502	0.892	0.895	0.897	0.899	0.901	0.900
Average	0.502	0.923	0.929	0.936	0.942	0.944	0.944