RESMARK: LATE-STAGE RESIDUAL INSERTION FOR FIDELITY-PRESERVING WATERMARKING IN DIFFUSION DENOISERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-image diffusion models need reliable watermarking for provenance and attribution. Denoiser-side watermarking couples the watermark to the iterative denoising process. A first-order sensitivity analysis suggests that early perturbations are transported through more subsequent solver transitions. As a result, the watermarked output can differ from its watermark-free counterpart under matched prompts and noise—a phenomenon we call paired-output drift. We formulate denoiser-side watermarking as late-stage residual insertion. Low-Noise Training (LNT) trains the watermark LoRA on low-noise timesteps. At inference, late-step activation enables it only over the final solver steps through a soft ramp. Because the first active step begins from a watermark-free trajectory, we propose a Teacher-Relative Insertion Loss (TRI), which trains the LoRA-enabled prediction to match the frozen base model prediction plus the desired watermark residual on the same watermark-free latent. Across SD1.5 U-Net and PixArt transformer denoisers, ResMark substantially improves paired-output fidelity over full-trajectory denoiser watermarking while retaining strong message recovery and robustness to conventional distortions and AI-regeneration attacks.

1 INTRODUCTION

Text-to-image diffusion models are now widely used to produce realistic visual content, increasing the need for reliable provenance and attribution (Rombach et al., 2022). An effective watermark should carry a recoverable message, remain detectable after common distortions, and preserve the output of the underlying generator (Zhu et al., 2018; Tancik et al., 2020; An et al., 2024). The last requirement concerns not only distribution-level image quality but also *paired-output fidelity*. Given the same prompt and initial noise, watermarking should not substantially change the semantic content or spatial composition of the corresponding watermark-free output.

Model-integrated diffusion watermarking methods differ in where the watermark is introduced. Decoder-side methods embed the watermark through the VAE decoder and leave the denoising trajectory unchanged (Fernandez et al., 2023; Rezaei et al., 2024; Ci et al., 2025). Denoiser-side methods instead couple it directly to the iterative prediction process, letting the watermark update participate throughout sampling (Feng et al., 2024). While decoder-side watermarking avoids trajectory drift by construction, it introduces the signal only during image decoding, where it may be overwritten by AI-based regeneration (Zhao et al., 2024; Liu et al., 2025). We therefore focus on the denoiser-side setting and seek to retain its regeneration robustness while recovering paired-output fidelity.

Full-trajectory activation lets a watermark LoRA modify early and intermediate denoising predictions, where global layout and object geometry are formed. A local first-order sensitivity analysis suggests that an earlier perturbation is transported through more subsequent solver transitions. Consequently, the watermark-enabled output can differ from the watermark-free output under the same prompt and initial noise, even when both remain realistic and prompt-consistent. We call this phenomenon *paired-output drift*. To reduce it, we formulate denoiser-side watermarking as late-stage residual insertion. We train the watermark LoRA with Low-Noise Training (LNT), which restricts training to low-noise timesteps. At inference, late-step activation keeps the LoRA disabled during the early

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.

solver steps, so the trajectory first follows the frozen base model, and enables it only over the final steps through a soft ramp. LNT and late-step activation are thus coupled. The inference gate confines the LoRA to the same low-noise regime used during training.

Late activation changes the input state at which watermark insertion is required. AquaLoRA's Prior-Preserving Fine-Tuning (PPFT) evaluates the LoRA-enabled denoiser on a residual-bearing noisy latent (Feng et al., 2024). This condition is compatible with full-trajectory sampling after early LoRA-induced deviations have accumulated. The same deviations, however, are also the source of paired-output drift. In contrast, the first active step of late activation begins from a watermark-free base trajectory. We therefore propose a Teacher-Relative Insertion Loss (TRI), which evaluates the LoRA-enabled and frozen base models on the same watermark-free latent and constrains their clean-latent prediction difference to match the target watermark residual.

Our contributions are summarized as follows:

- We identify paired-output drift in full-trajectory denoiser-side watermarking and provide a local first-order sensitivity analysis showing why early perturbations traverse more subsequent solver transitions. To limit this effect, we combine Low-Noise Training (LNT) with late-step soft activation, which confines watermark intervention to the low-noise tail of sampling.
- We identify the mismatch between residual-bearing training inputs and the watermark-free input at the onset of late-step activation. We address this mismatch with Teacher-Relative Insertion Loss (TRI), which applies teacher-relative residual matching to message-conditioned LoRA within the LNT interval, using watermark-free forward-noised latents as proxies for the activation-boundary input.
- We evaluate ResMark through unified comparisons with prior methods on SD1.5, controlled component analyses on PixArt-α, and evaluations across U-Net and transformer denoisers. ResMark improves paired-output fidelity while maintaining strong message recovery and robustness to conventional distortions and AI-regeneration attacks.

2 Related Work

2.1 Traditional Image Watermarking

Traditional image watermarking is model-agnostic and typically modifies an already rendered image using transform-domain or spread-spectrum techniques (Cox et al., 1997; Kundur & Hatzinakos, 1998). Neural methods instead learn an encoder–decoder pair end to end. HiDDeN trains through differentiable distortions to recover embedded bit strings (Zhu et al., 2018), while StegaStamp targets distortions introduced by printing and photography (Tancik et al., 2020). Because these methods operate after rendering, their provenance guarantees depend on consistently applying the watermark encoder and are not intrinsically coupled to the generator's parameters or sampling process.

2.2 Watermarking of Diffusion Models

Diffusion watermarking includes sampling-side and model-integrated approaches. We further distinguish model-integrated methods according to whether the watermark is introduced through the decoder or the denoising network.

Sampling-side watermarking. Sampling-side methods introduce the watermark through variables used during generation while leaving the model parameters unchanged (Zhang et al., 2025). Tree-Ring embeds a structured pattern in the Fourier domain of the initial noise and detects it through diffusion inversion (Wen et al., 2023), whereas Gaussian Shading encodes messages through the latent noise distribution (Yang et al., 2024). Recent methods further exploit the temporal structure of sampling (Luo et al., 2026). ROBIN inserts a watermark into an intermediate diffusion state and uses optimized prompt guidance over the remaining denoising steps (Huang et al., 2024), while OptMark places complementary watermark signals at different sampling times through inference-time optimization (Xing et al., 2026). These approaches can be applied to frozen models, but their embedding and detection procedures depend on manipulating designated sampling states, initial latents, or inference guidance.

Decoder-side watermarking. Decoder-side methods introduce watermark information through the VAE decoder after latent denoising. Stable Signature fine-tunes a decoder copy to embed a fixed binary signature (Fernandez et al., 2023). LaWa and WMAdapter attach message-conditioned residual modules to decoder features (Rezaei et al., 2024; Ci et al., 2025), while EW-LoRA applies low-rank updates to selected VAE decoder layers (Lin et al., 2026). StableGuard equips a frozen VAE decoder with residual watermark adapters and jointly learns a forensic network for watermark verification and tamper localization (Yang et al., 2025). MOLM uses key-dependent routing among LoRA adapters. Its default configuration routes adapters through VAE-decoder ResNet blocks, whereas a higher-capacity U-Net variant incurs noticeable generation-quality degradation (Fares et al., 2026). These decoder-side configurations introduce the watermark only during image decoding and therefore leave the iterative latent denoising trajectory unchanged.

Denoiser-side watermarking. Denoiser-side methods encode the watermark in the noise-prediction network, coupling watermark insertion to iterative generation (Wang et al., 2025). AquaLoRA learns a message-conditioned Watermark LoRA for the Stable Diffusion U-Net using Prior-Preserving Fine-Tuning (PPFT) (Feng et al., 2024). AuthenLoRA jointly trains a stylization LoRA and watermark embedding for a target LoRA asset (Shi et al., 2025). LoRA-Key learns a standalone, user-specific Watermark LoRA that can be composed with independently trained target LoRAs through training-free linear superposition (Wang et al., 2026). To our knowledge, prior denoiser-side watermarking methods do not explicitly study time-restricted activation of a message-conditioned Watermark LoRA or its matched-seed paired-output fidelity. Outside watermarking, T-LoRA uses timestep-dependent rank masking for diffusion personalization (Soboleva et al., 2026), while TimeStep Master trains and combines timestep-interval LoRA experts for diffusion adaptation (Zhuang et al., 2025). Both target customization or adaptation rather than watermark insertion or matched-seed paired-output fidelity.

Our work retains AquaLoRA's message-conditioned denoiser LoRA, pretrained watermark encoder and decoder, and compositional message mapper (Feng et al., 2024). SleeperMark (Wang et al., 2025) uses teacher-relative latent residual matching with timestep-dependent loss weights to learn a fixed watermark activated by a text trigger, directly fine-tuning selected denoiser parameters. ResMark combines per-generation message conditioning without a text trigger, LNT, and explicit late-step LoRA activation on a frozen backbone. The zero gate preserves the base trajectory before activation, while TRI learns insertion from a watermark-free proxy for the activation boundary. We study this coupling through matched-seed paired-output fidelity.

3 Proposed Method

Figure 1 gives an overview of the proposed framework, consisting of (a) training with LNT and the Teacher-Relative Insertion Loss (TRI), and (b) inference with late-step activation.

3.1 Problem Setup

We consider a latent text-to-image diffusion model with a frozen VAE encoder–decoder pair $(\mathcal{E}, \mathcal{D})$ and a frozen text-conditioned denoiser ϵ_{θ_0} (Rombach et al., 2022). Given an image x, let $z_0 = \mathcal{E}(x)$ denote its scaled VAE latent, and let z_t denote its standard forward-noised latent at timestep t (Ho et al., 2020). We write $\widehat{z}_0(z_t, t, \widehat{\epsilon})$ for the corresponding clean-latent estimate obtained from a noise prediction $\widehat{\epsilon}$. All latent variables are expressed in the scaled coordinates of the pretrained VAE. The explicit forward process and clean-latent conversion are given in Appendix A.

Our goal is to train a message-conditioned denoiser LoRA while keeping the base denoiser and VAE frozen. Given a payload $m \in \{0,1\}^L$ with $L = 48$, we optimize the LoRA parameters ψ and message-mapper parameters ϕ so that a frozen watermark decoder can recover m from the resulting image, while the watermarked output remains close to the frozen base output under the same prompt and initial noise.

We reuse the pretrained latent watermark encoder E_{sec} and its paired image-level decoder D_{sec} released with AquaLoRA (Feng et al., 2024). We absorb the fixed VAE scaling into E_{sec} and denote its cover-agnostic residual in the scaled latent coordinates by $\Delta(m) = E_{\text{sec}}(m)$. This residual specifies the desired watermark-induced shift in clean-latent space. The decoder D_{sec} extracts the payload from the final RGB image, and neither E_{sec} nor D_{sec} receives gradients during training.

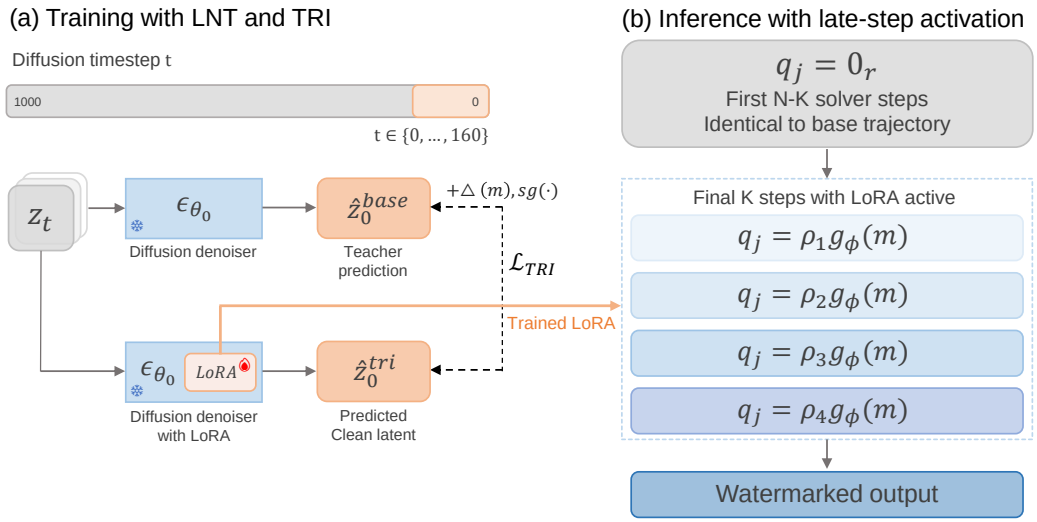

Figure 1: **Overview.** (a) Training. Timesteps are sampled from the low-noise window used by LNT. The frozen base and LoRA-enabled denoisers receive the same watermark-free z_t. TRI matches the LoRA-enabled clean-latent estimate to $\mathrm{sg}(\hat{z}_0^{\mathrm{base}} + \Delta(m))$. Snowflakes denote frozen modules. Flames denote trainable LoRA parameters. (b) Inference. The LoRA is disabled for the first $N - K$ solver steps and activated over the final K steps through a monotone ramp. Exact backbone-specific schedule parameters are provided in Appendix A.3.

3.2 Message-Conditioned LoRA Parameterization

Following AquaLoRA (Feng et al., 2024), a compositional message mapper converts a payload into a rank-wise modulation vector $q(m) = g_\phi(m) \in \mathbb{R}^r$. This vector modulates the low-rank updates of the denoiser LoRA (Hu et al., 2022). The complete layerwise parameterization is provided in Appendix A. We collect the trainable LoRA factors as ψ and jointly optimize ψ and ϕ.

At full watermark strength, we use $q = g_\phi(m)$. When $q = \mathbf{0}_r$, every low-rank update vanishes and the LoRA-adapted denoiser exactly recovers the frozen base denoiser. This property enables the runtime control in Section 3.3: the sampler follows the base trajectory while $q_j = \mathbf{0}_r$ and introduces the message-conditioned update only during the selected late steps.

We assume an owner-controlled generation pipeline that invokes the prescribed watermark-enabled sampler. Resistance to deliberate LoRA deactivation or white-box removal is outside our threat model.

3.3 LNT: Low-Noise Training for Late-Step Activation

When the watermark LoRA is active throughout sampling, its updates at high and intermediate noise can propagate through subsequent solver steps. These propagated perturbations can alter content and spatial layout, producing paired-output drift even under matched prompts and initial noise. Since a watermark serves as a provenance signal rather than an additional semantic condition, we restrict its intervention to the low-noise tail of the denoising process, as illustrated in Figure 1(b).

Motivation for late-step activation. We analyze how a LoRA-induced prediction perturbation affects the final sample. For a differentiable reverse solver, let u_i denote the direct LoRA-induced prediction perturbation at step i, G_i the local sensitivity of the solver update to that perturbation, and J_ℓ the local Jacobian of each subsequent base-model transition. A first-order expansion yields

$$d_{N+1} \approx \sum_{i=1}^{N} \underbrace{J_N J_{N-1} \cdots J_{i+1}}_{\text{subsequent solver transitions}} G_i u_i,$$

where d_{N+1} is the final deviation from the LoRA-off trajectory. Thus, an earlier perturbation is transported through more subsequent solver transitions. This local sensitivity analysis motivates delaying watermark intervention to the final denoising steps. The full derivation, including the extension to multistep solvers, is provided in Appendix B.

Low-noise training. LNT samples diffusion timesteps uniformly from a discrete low-noise interval rather than from the full diffusion schedule. For watermark-enabled denoiser evaluations during training, we use the full modulation $q = g_\phi(m)$ at the sampled timestep. We select this interval to cover the low-noise time levels associated with the tail of the chosen inference schedule. This specializes the watermark LoRA to the low-noise regime used at inference. Although forward-noised training latents do not exactly reproduce solver trajectory states, they provide a tractable low-noise proxy for this regime. The inference gate is coupled to LNT and confines the LoRA to this same regime.

Late-step activation. Let an N-step solver visit diffusion time levels $\tau_1 > \tau_2 > \cdots > \tau_N$, ordered from high to low noise, and let K denote the number of final solver steps used for watermark insertion. At solver step $j \in \{1, \ldots, N\}$, the rank-wise modulation vector supplied to the LoRA branches is

$$q_j(m) = \begin{cases} \mathbf{0}_r, & 1 \le j \le N-K, \\ \rho_{j-(N-K)}\, g_\phi(m), & N-K < j \le N. \end{cases} \tag{1}$$

where $0 < \rho_1 \le \cdots \le \rho_K \le 1$ defines a monotone ramp. The terminal strength ρ_K is selected for each backbone and reported in Table 5. During the first $N-K$ steps, $q_j(m) = \mathbf{0}_r$, so every denoiser evaluation is identical to that of the frozen base denoiser. Equivalently, this gate sets $u_i = 0$ for $i \le N-K$ in the first-order expansion above, removing the corresponding early terms. When classifier-free guidance (Ho & Salimans, 2022) is used, we apply the same timestep gate to both the conditional and unconditional denoiser evaluations, so the guided prediction is also identical to the base model during the first $N-K$ steps. Given the same prompt, initial noise, solver, and coupled random draws for a stochastic solver, the watermark-enabled and base trajectories are therefore identical through the first $N-K$ solver updates. The final K steps progressively activate the message-conditioned LoRA, while the ramp avoids a full-strength update at the first active step. A hard late gate is recovered by setting $\rho_k = 1$ for every $k \in \{1, \ldots, K\}$. The fractional ramp values are not separately trained message states. They scale the direct low-rank update in each adapted layer, while the overall denoiser remains nonlinear. We therefore treat the ramp as an inference schedule and evaluate it empirically against hard activation.

3.4 Teacher-Relative Insertion Loss

Late activation begins from a solver state produced by the LoRA-off sampler. We therefore use a watermark-free forward-noised latent as a tractable proxy for the activation-boundary condition. For an image–caption pair (x, c), an independently sampled payload m, and a timestep t sampled from the LNT low-noise interval, let z_t denote the corresponding watermark-free noisy latent. The frozen base model and the LoRA-enabled model receive the same (z_t, t, c), producing clean-latent estimates $\widehat{z}_0^{\text{base}}$ and $\widehat{z}_0^{\text{tri}}$, respectively. The LoRA-enabled model uses the full message modulation $q = g_\phi(m)$.

We optimize the Teacher-Relative Insertion Loss

$$\mathcal{L}_{\text{TRI}}(\psi, \phi) = \mathbb{E}\left[\text{MSE}\left(\widehat{z}_0^{\text{tri}}, \text{sg}\left(\widehat{z}_0^{\text{base}} + \Delta(m)\right)\right)\right]. \tag{2}$$

The expectation is taken over training pairs, independently sampled payloads, timesteps, and forward noise. Here, $\text{sg}(\cdot)$ denotes stop-gradient, applied to the entire teacher-relative target. TRI shares its teacher-relative residual-matching form with the watermark-insertion term of SleeperMark (Wang et al., 2025). It constrains the LoRA-induced clean-latent shift $\widehat{z}_0^{\text{tri}} - \widehat{z}_0^{\text{base}}$ to match the target watermark residual $\Delta(m)$. Here, we apply this supervision to message-conditioned LoRA within the LNT interval, with explicit late-step activation at inference.

AquaLoRA-style PPFT (Feng et al., 2024) evaluates the LoRA-enabled denoiser on a residual-bearing noisy latent. TRI instead evaluates the LoRA-enabled and frozen base denoisers on the same watermark-free z_t and applies a teacher-relative target. It therefore aligns training with the watermark-free input-state condition induced by late-step activation.

TRI supervises a relative clean-latent prediction rather than explicitly adding $\Delta(m)$ to the solver state. The final K solver steps therefore do not correspond to K additive watermark injections.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Implementation. We evaluate ResMark at 512×512 resolution on Stable Diffusion v1.5 (SD1.5) (Rombach et al., 2022) and PixArt-α XL/2 (Chen et al., 2024). For both backbones, we freeze the base denoiser, VAE, and pretrained watermark encoder and decoder. We optimize only the message-conditioned LoRA parameters and message mapper using the Teacher-Relative Insertion Loss with Low-Noise Training.

For each backbone, the training set contains 20,000 image–caption pairs. It comprises 10,000 images randomly sampled from the COCO2017 training split with their associated captions (Lin et al., 2014) and 10,000 images generated by the corresponding frozen base sampler using prompts from the Gustavosta Stable-Diffusion-Prompts dataset (Gustavosta, 2022). We sample an independent 48-bit payload for each training example. All training runs are performed on an NVIDIA RTX A6000 GPU. We use backbone-specific LoRA target modules, ranks, and late-step activation schedules. Detailed training and inference configurations are provided in Appendix A.3.

Baselines and evaluation. For the same-backbone comparison on SD1.5, we evaluate Stable Signature, LaWa, MOLM, and AquaLoRA (Fernandez et al., 2023; Rezaei et al., 2024; Fares et al., 2026; Feng et al., 2024) using their released implementations and reported watermarking procedures. StableGuard is evaluated using its officially released pretrained checkpoint based on SD2.1-base (Yang et al., 2025). We additionally train and evaluate ResMark on PixArt-α XL/2 to assess its applicability to a transformer denoiser. The SD1.5 rows therefore provide a same-backbone comparison of complete watermarking systems, while the StableGuard and PixArt rows are interpreted within their respective backbones. We isolate the effects of LNT and TRI through controlled ablations on a common PixArt backbone with identical evaluation and sampling settings.

For each method, we generate 1,000 images from 100 prompts selected from PartiPrompts (Yu et al., 2022) and 10 initial-noise seeds per prompt. These evaluation prompts are disjoint from the COCO2017 training captions and Gustavosta Stable-Diffusion-Prompts used to train ResMark. On clean generated images, we report bit accuracy and TPR using a payload-length-specific binomial threshold corresponding to a nominal FPR on the order of 10^{-6}, together with paired PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018). For distortion robustness, we report distortion-wise bit accuracy. All bit-accuracy values are reported in percent. Separately, we report $\Delta\mathrm{FID}$ using the FID metric (Heusel et al., 2017) and the 5,000-image protocol detailed in Appendix C.

4.2 MAIN RESULTS

Table 1 summarizes the system-level comparison. The SD1.5 block provides the primary same-backbone comparison. Within the denoiser-side setting, ResMark improves both paired-output fidelity and watermark detectability over AquaLoRA. ResMark increases bit accuracy from 95.42% to 97.00% and TPR from 0.990 to 0.994. It also improves paired fidelity substantially, increasing PSNR from 17.39 dB to 32.10 dB and SSIM from 0.652 to 0.961, while reducing LPIPS from 0.2915 to 0.0266. The same comparison reduces $\Delta\mathrm{FID}$ from 0.573 to 0.386.

These results support the central premise of ResMark. Watermark intervention can be confined to the low-noise tail of sampling while retaining reliable payload recovery and substantially reducing deviation from the matched watermark-free output. The decoder-side methods serve as broader system-level references rather than controlled denoiser-side ablations.

StableGuard is interpreted within its released SD2.1-base backbone. The PixArt-α XL/2 result further demonstrates the applicability of ResMark to a transformer denoiser beyond the SD1.5 U-Net setting. It is reported as a separate-backbone result and is excluded from the bold and underline ranking. Since $\Delta\mathrm{FID}$ is computed relative to the watermark-free generator within each backbone, we use it as a distributional diagnostic rather than as a cross-backbone ranking metric.

Table 1: Main results on clean generated images. Stable Signature, LaWa, MOLM, AquaLoRA, and ResMark are reproduced on a common SD1.5 backbone. The PixArt-α XL/2 row evaluates ResMark on a transformer denoiser. MOLM uses a 28-bit payload, whereas the other methods use 48 bits. For the detection and paired-fidelity metrics, bold and underlined values denote the best and second-best results, respectively, among all evaluated methods except the ResMark PixArt-α XL/2 transformer-backbone result. ΔFID is computed with the separate 5,000-image COCO2017 validation protocol and is interpreted only within a backbone.

Method	**Backbone**	**Bits**	**Detection**		**Paired fidelity**			ΔFID
			Bit-Acc.↑	TPR↑	PSNR↑	SSIM↑	LPIPS↓	
Decoder-side								
Stable Signature	SD1.5	48	<u>99.04</u>	<u>0.998</u>	27.96	0.862	0.0452	−0.276
LaWa	SD1.5	48	**100.00**	**1.000**	29.05	0.850	0.0422	−0.621
StableGuard	SD2.1-base	48	75.96	0.263	26.74	0.824	0.0783	−0.559
MOLM	SD1.5	28	89.10	0.371	<u>31.65</u>	**0.976**	**0.0174**	−0.224
Denoiser-side								
AquaLoRA	SD1.5	48	95.42	0.990	17.39	0.652	0.2915	0.573
ResMark (Ours)	SD1.5	48	97.00	0.994	**32.10**	<u>0.961</u>	<u>0.0266</u>	0.386
ResMark (Ours)	PixArt-α XL/2	48	98.39	1.000	32.69	0.967	0.0232	−0.009

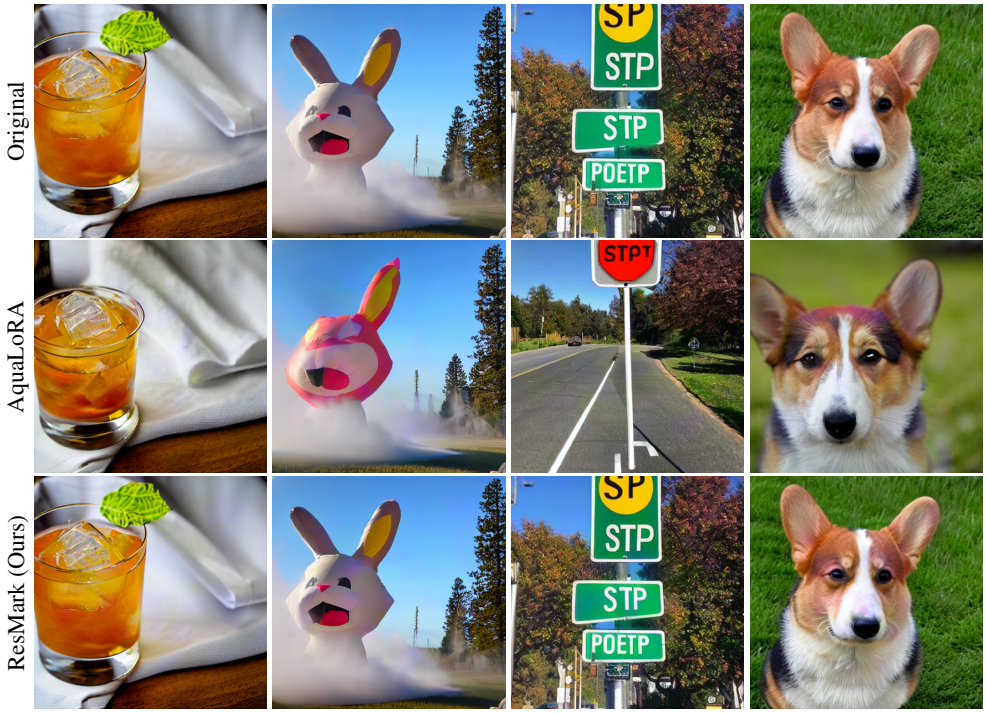

Figure 2: **Controlled qualitative comparison on SD1.5.** Each column uses an identical prompt, initial-noise seed, solver, and sampling configuration. The top row shows the watermark-free reference generated by the frozen base sampler with the LoRA disabled. The middle row shows AquaLoRA with full-trajectory activation. The bottom row shows ResMark with LNT, TRI, and late soft activation. AquaLoRA can alter global layout or object identity relative to the reference, whereas ResMark preserves the corresponding watermark-free output while embedding the payload.

Figure 2 visualizes the same effect on SD1.5 under matched prompts and initial noise. AquaLoRA changes layout or object composition in several examples, whereas ResMark preserves the corresponding watermark-free output while embedding the payload.

Table 2: Robustness under 7 signal-processing distortions and 3 AI-regeneration attacks. Bold and underlined values denote the best and second-best results among all evaluated methods except the ResMark PixArt-α XL/2 transformer-backbone result, respectively.

Method	Signal processing							AI regeneration		
	Color jitter	Blur	Noise	JPEG	Crop	BM3D	Resize	SDEdit	Rinse-2	VAE-B
Decoder-side										
Stable Signature	<u>97.95</u>	89.30	94.64	87.55	<u>98.83</u>	88.04	81.65	53.56	52.26	78.99
LaWa	**100.00**	**98.85**	**99.61**	**99.83**	**99.86**	**97.22**	**99.45**	76.30	70.00	90.34
StableGuard	79.08	63.58	77.31	70.81	60.48	55.00	61.93	52.36	52.36	54.32
MOLM	85.36	88.74	89.28	84.39	88.77	86.29	86.30	69.03	65.15	81.13
Denoiser-side										
AquaLoRA	92.67	95.39	94.09	93.89	92.17	94.78	95.31	<u>85.73</u>	<u>76.55</u>	<u>92.74</u>
ResMark (Ours) (SD1.5)	94.67	<u>97.00</u>	<u>95.69</u>	<u>95.96</u>	92.78	<u>96.43</u>	<u>96.93</u>	**86.58**	**76.64**	**93.56**
ResMark (Ours) (PixArt)	96.18	98.38	97.42	97.44	95.64	97.96	98.28	90.65	81.84	96.38

4.3 Distortion Robustness

We evaluate watermark detection after seven signal-processing distortions and three AI-regeneration transformations. Exact transformation settings are provided in Appendix C.

Table 2 reports distortion-wise bit accuracy. On the common SD1.5 backbone, ResMark improves over AquaLoRA under all 10 transformations. It achieves bit accuracy from 92.78% to 97.00% across the 7 signal-processing distortions, whereas AquaLoRA ranges from 92.17% to 95.39%. LaWa remains strongest under most conventional signal-processing distortions, while ResMark is second-best on blur, noise, JPEG, BM3D, and resize among the ranked methods.

Under AI regeneration, ResMark achieves the highest ranked bit accuracy on SDEdit, Rinse-2, and VAE-B, reaching 86.58%, 76.64%, and 93.56%, respectively. These results exceed AquaLoRA by 0.85, 0.09, and 0.82 percentage points. Although these margins are modest, the improvement is consistent across all 3 transformations.

4.4 Ablation Studies

Progressive component analysis. Table 3 examines late activation, LNT, and TRI on PixArt. Applying late soft activation to the full-timestep PPFT checkpoint improves paired fidelity but weakens detection. Restricting PPFT training to low-noise timesteps further improves fidelity yet reduces detection. Replacing PPFT with TRI under the same low-noise and late-soft configuration increases bit accuracy from 75.58% to 98.39% and TPR from 0.233 to 1.000. Although this trades some fidelity for reliable insertion, the complete ResMark configuration retains substantially better paired fidelity than full-trajectory PPFT: PSNR increases from 18.06 to 32.69 dB, SSIM from 0.701 to 0.967, and LPIPS decreases from 0.2528 to 0.0232.

With TRI and the late-soft gate fixed, LNT increases bit accuracy from 88.14% to 98.39% and TPR from 0.837 to 1.000 while maintaining comparable paired fidelity (PSNR: 32.59 vs. 32.69 dB; LPIPS: 0.0233 vs. 0.0232). Thus, timestep restriction improves detection under TRI but reduces it under PPFT, supporting the LNT–TRI coupling for reliable late-step insertion with high paired-output fidelity in the evaluated setting.

Activation schedule. Table 4 compares hard and soft late activation using the same final PixArt LNT–TRI checkpoint. Both schedules activate the LoRA over the final $K = 4$ solver steps. The hard gate achieves slightly higher bit accuracy, while both schedules reach a TPR of 1.000. The soft ramp substantially improves paired fidelity, increasing PSNR from 28.80 to 32.69 dB and SSIM from 0.937 to 0.967 while reducing LPIPS from 0.0562 to 0.0232. We therefore adopt the soft ramp as the more favorable detection–fidelity operating point.

Table 3: Controlled component analysis on PixArt. Each configuration is written as training timestep range / objective / inference activation, and all late-soft schedules use $K = 4$. The first two rows use the same full-timestep PPFT checkpoint. TRI denotes the Teacher-Relative Insertion Loss. The last two rows isolate the effect of Low-Noise Training under a fixed TRI objective and a fixed late-soft gate. Paired metrics are computed against matched LoRA-off outputs. Bit accuracy is reported in percent.

Configuration	**Detection**		**Paired fidelity**		
	Bit-Acc.↑	TPR↑	PSNR↑	SSIM↑	LPIPS↓
Full / PPFT / Full	94.45	0.968	18.06	0.701	0.2528
Full / PPFT / Late soft	86.35	0.709	33.64	0.968	0.0164
Low-noise / PPFT / Late soft	75.58	0.233	38.02	0.986	0.0058
Full / TRI / Late soft	88.14	0.837	32.59	0.962	0.0233
Low-noise / TRI / Late soft (Ours)	**98.39**	**1.000**	32.69	0.967	0.0232

Table 4: Effect of the late-step activation schedule on PixArt using the same final LNT–TRI checkpoint and a $K = 4$ activation window. The hard gate uses $\rho = [1, 1, 1, 1]$, while the soft ramp uses $\rho = [0.25, 0.5, 0.75, 1.0]$. The best result for each metric is shown in bold.

Schedule	**Detection**		**Paired fidelity**		
	Bit-Acc.↑	TPR↑	PSNR↑	SSIM↑	LPIPS↓
Hard gate	**99.14**	**1.000**	28.80	0.937	0.0562
Soft ramp (Ours)	98.39	**1.000**	**32.69**	**0.967**	**0.0232**

LoRA rank. The LoRA rank controls the capacity available for message-conditioned residual insertion. Based on the PixArt capacity sweep, we select rank 128, which achieves the strongest detection while retaining high paired fidelity. Detailed results are reported in Table 8.

4.5 Limitations

Some generated images exhibit a low-frequency color pattern. Our qualitative inspection finds a similar pattern when the frozen watermark residual $\Delta(m)$ is added directly to a watermark-free VAE latent before decoding. This suggests that the artifact may originate from the pretrained watermark encoder rather than from late-step activation or TRI. Because the watermark encoder and decoder remain frozen, ResMark inherits this limitation. Mitigating the artifact may require retraining E_{sec} and D_{sec} with explicit color-preservation or perceptual regularization.

Our evaluation covers an SD1.5 U-Net and a PixArt transformer denoiser, while the controlled component analyses focus on PixArt. Evaluation across additional denoiser architectures, image resolutions, and sampling schedules remains future work. ResMark also assumes an owner-controlled generation pipeline that invokes the prescribed watermark-enabled sampler. It does not address deliberate LoRA deactivation, white-box watermark removal, or adaptive attacks targeting the watermark decoder.

5 Conclusion

We have presented ResMark, a late-stage residual insertion framework for denoiser-side multi-bit watermarking. LNT specializes the watermark LoRA to the low-noise regime, while late-step activation keeps the early sampling trajectory aligned with the frozen base model. TRI then learns watermark insertion from a watermark-free input state by matching the LoRA-induced clean-latent shift to the target watermark residual relative to the frozen base prediction. Experiments on SD1.5 and PixArt show that ResMark substantially improves paired-output fidelity over full-trajectory denoiser watermarking while retaining strong message recovery and robustness to the conventional distortions and AI-regeneration attacks. Future work will improve the underlying watermark encoder and decoder and evaluate late-stage insertion across a broader range of diffusion architectures and attack settings.

AI USE STATEMENT

We used generative AI tools to assist with literature discovery and retrieval, to support code implementation, to draft portions of the manuscript, and to improve the clarity and readability of the writing. All AI-assisted text and identified references were reviewed and verified by the authors. The authors take full responsibility for the content of this paper.

REFERENCES

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 *Proceedings of the 41st 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.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International Conference on Learning Representations*, 2024.

Hai Ci, Yiren Song, Pei Yang, Jinheng Xie, and Mike Zheng Shou. WMAdapter: Adding watermark control to latent diffusion models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10901–10919. PMLR, 2025.

Ingemar J. Cox, Joe Kilian, F. Thomson Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, 1997. doi: 10.1109/83.650120.

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.

Samar Fares, Nurbek Tastan, Noor Hazim Hussein, and Karthik Nandakumar. MOLM: Mixture of LoRA markers. In *International Conference on Learning Representations*, 2026.

Weitao Feng, Wenbo Zhou, Jiyan He, Jie Zhang, Tianyi Wei, Guanlin Li, Tianwei Zhang, Weiming Zhang, and Nenghai Yu. AquaLoRA: Toward white-box protection for customized stable diffusion models via watermark LoRA. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 13423–13444. PMLR, 2024.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22466–22477, 2023. doi: 10.1109/ICCV51070.2023.02053.

Gustavosta. Stable-Diffusion-Prompts. Hugging Face dataset, 2022.

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*, volume 30, pp. 6626–6637, 2017.

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

Jonathan Ho, Ajay N. Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

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

Huayang Huang, Yu Wu, and Qian Wang. ROBIN: Robust and invisible watermarks for diffusion models with adversarial optimization. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0129.

Deepa Kundur and Dimitrios Hatzinakos. Digital watermarking using multiresolution wavelet decomposition. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, pp. 2969–2972, 1998. doi: 10.1109/ICASSP.1998.678149.

Dongdong Lin, Yue Li, Benedetta Tondi, Kaiqing Lin, Bin Li, and Mauro Barni. An efficient watermarking method for latent diffusion models via low-rank adaptation and dynamic loss weighting. *Expert Systems with Applications*, 331:133172, 2026. doi: 10.1016/j.eswa.2026.133172.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, volume 8693 of *Lecture Notes in Computer Science*, pp. 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48.

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

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.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22 (4):730–751, 2025. doi: 10.1007/s11633-025-1562-4.

Pengcheng Luo, Zexi Jia, Yijia Zhong, Jinchao Zhang, and Jie Zhou. GROW: Watermark generation with progressive guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Ahmad Rezaei, Mohammad Akbari, Saeed Ranjbar Alvar, Arezou Fatemi, and Yong Zhang. LaWa: Using latent space for in-generation image watermarking. In *Computer Vision – ECCV 2024*, volume 15147 of *Lecture Notes in Computer Science*, pp. 118–136. Springer, 2024. doi: 10.1007/978-3-031-73024-5_8.

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.

Fangming Shi, Li Li, Kejiang Chen, Guorui Feng, and Xinpeng Zhang. AuthenLoRA: Entangling stylization with imperceptible watermarks for copyright-secure LoRA adapters. *arXiv preprint arXiv:2511.21216*, 2025. doi: 10.48550/arXiv.2511.21216.

Vera Soboleva, Aibek Alanov, Andrey Kuznetsov, and Konstantin Sobolev. T-LoRA: Single image diffusion model customization without overfitting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(11):9051–9059, 2026. doi: 10.1609/aaai.v40i11.37861.

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

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*, 2020. doi: 10.1109/CVPR42600.2020.00219.

Yaopeng Wang, Qingliang Wang, Zhibo Wang, Huiyu Xu, Jiacheng Du, Qiu Wang, Jia-Li Yin, and Kui Ren. LoRA-Key: User-centric LoRA watermarking for text-to-image diffusion models. *arXiv preprint arXiv:2605.29569*, 2026. doi: 10.48550/arXiv.2605.29569.

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. doi: 10.1109/TIP.2003.819861.

Zilan Wang, Junfeng Guo, Jiacheng Zhu, Yiming Li, Heng Huang, Muhao Chen, and Zhengzhong Tu. SleeperMark: Towards robust watermark against fine-tuning text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8213–8224, 2025. doi: 10.1109/CVPR52734.2025.00769.

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, 2023. doi: 10.52202/075280-2529.

Jiazheng Xing, Hai Ci, Hongbin Xu, Hangjie Yuan, Yong Liu, and Mike Zheng Shou. OptMark: Robust multi-bit diffusion watermarking via inference time optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(13):11150–11158, 2026. doi: 10.1609/aaai.v40i13.38094.

Haoxin Yang, Bangzhen Liu, Xuemiao Xu, Cheng Xu, Yuyang Yu, Zikai Huang, Yi Wang, and Shengfeng He. StableGuard: Towards unified copyright protection and tamper localization in latent diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0495.

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.

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

Guokai Zhang, Lanjun Wang, Yuting Su, and An-An Liu. MarkPlugger: Generalizable watermark framework for latent diffusion models without retraining. *IEEE Transactions on Multimedia*, 2025. doi: 10.1109/TMM.2025.3565960.

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.

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, 2024. doi: 10.52202/079017-0276.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. HiDDeN: Hiding data with deep networks. In *Computer Vision – ECCV 2018*, volume 11219 of *Lecture Notes in Computer Science*, pp. 682–697. Springer, 2018. doi: 10.1007/978-3-030-01267-0_40.

Shaobin Zhuang, Yiwei Guo, Yanbo Ding, Kunchang Li, Xinyuan Chen, Yaohui Wang, Fangyikang Wang, Ying Zhang, Chen Li, and Yali Wang. TimeStep Master: Asymmetrical mixture of timestep LoRA experts for versatile and efficient diffusion models in vision. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 80499–80515. PMLR, 2025.

A Diffusion and Message-Conditioned LoRA Details

A.1 Latent Diffusion Notation

For an image x, the frozen VAE encoder of the latent diffusion model (Rombach et al., 2022) produces the scaled latent

$$z_0 = \mathcal{E}(x). \tag{3}$$

Here, $\mathcal{E}$ includes the fixed latent scaling of the pretrained VAE. Under the forward diffusion process (Ho et al., 2020) with cumulative noise schedule $\bar{\alpha}_t$, the noisy latent at timestep t is

$$z_t = \sqrt{\bar{\alpha}_t}\, z_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{4}$$

Given a noise prediction $\widehat{\epsilon}$, the corresponding clean-latent estimate is

$$\widehat{z}_0 = \frac{z_t - \sqrt{1-\bar{\alpha}_t}\,\widehat{\epsilon}}{\sqrt{\bar{\alpha}_t}}. \tag{5}$$

We use the pretrained latent watermark encoder E_{sec} released with AquaLoRA (Feng et al., 2024) in the scaled coordinates of the pretrained VAE. For a payload m, it produces the cover-agnostic residual $\Delta(m) = E_{\mathrm{sec}}(m)$. The nominal watermarked latent is

$$z_0^{\mathrm{wm}} = z_0 + \Delta(m). \tag{6}$$

This nominal latent defines the desired watermark residual. TRI instead uses the teacher-relative target $\widehat{z}_0^{\mathrm{base}} + \Delta(m)$ rather than directly regressing to z_0^{wm}.

A.2 Message-Conditioned LoRA

We follow the standard LoRA parameterization (Hu et al., 2022) and the message-conditioned formulation of AquaLoRA (Feng et al., 2024). For an adapted linear layer $\ell \in \mathcal{L}$ with frozen weight $W_\ell \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}$, let $A_\ell \in \mathbb{R}^{r \times d_{\mathrm{in}}}$ and $B_\ell \in \mathbb{R}^{d_{\mathrm{out}} \times r}$ denote its trainable rank-r LoRA factors. Omitting the frozen bias term for clarity, the message-conditioned layer computes

$$f_{\psi,\ell}(h; q) = W_\ell h + \frac{\alpha}{r} B_\ell \left(q \odot A_\ell h\right), \tag{7}$$

where h is the input activation, α is the LoRA scaling coefficient, and $\odot$ denotes element-wise multiplication. The modulation vector q is shared across adapted layers, while each layer has its own LoRA factors. For the transformer linear layers in PixArt, q is broadcast over token positions. For the convolutional targets in the SD1.5 U-Net, we use the analogous factorized convolutional operators and broadcast q over spatial locations. We collect the trainable factors as $\psi = \{(A_\ell, B_\ell)\}_{\ell \in \mathcal{L}}$.

Because the LoRA branches have no bias, setting $q = \mathbf{0}_r$ removes every low-rank update and exactly recovers the frozen base denoiser.

A.3 Backbone-Specific Training and Sampling Configurations

Prompts for the synthetic training images are drawn from the Gustavosta Stable-Diffusion-Prompts dataset, available at `https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts`.

We use backbone-specific LoRA target modules, ranks, and late-step activation schedules. Table 5 summarizes the complete training and sampling configurations. For PixArt-α XL/2 (Chen et al., 2024), the denoiser predicts both noise and auxiliary variance channels. We use only the noise-prediction channels to construct $\widehat{z}_0^{\mathrm{base}}$ and $\widehat{z}_0^{\mathrm{tri}}$ and to compute TRI. The auxiliary variance channels are excluded from the loss. Both backbone configurations use DPM-Solver++ (2M) (Lu et al., 2025). The VAE, pretrained watermark encoder, and pretrained watermark decoder remain frozen for both backbones.

For the full-timestep and low-noise TRI configurations in Table 3, we use the same training data, LoRA rank, optimization settings, and number of training updates; only the sampled timestep range differs.

Table 5: Backbone-specific training and sampling configurations.

Setting	**SD1.5**	**PixArt-α XL/2**
Resolution	512×512	512×512
LoRA target modules	All 256 LoRA-compatible modules (192 attention/FF linear + 64 ResNet conv)	All linear modules (286 layers)
LoRA rank r	256	128
LoRA scaling α	256	128
Payload length	48 bits	48 bits
Batch size	4	4
Training epochs	5 (25,000 steps)	5 (25,000 steps)
Optimizer and weight decay	AdamW, 0.01	AdamW, 0.01
Learning-rate schedule	Cosine with warmup (500 steps)	Cosine with warmup (500 steps)
Peak learning rate	1.5×10^{-4}	1.5×10^{-4}
Minimum learning rate	7×10^{-5}	7×10^{-5}
LNT timestep interval	$\{0, \ldots, 160\}$	$\{0, \ldots, 160\}$
Solver	DPM-Solver++ (2M)	DPM-Solver++ (2M)
Solver steps N	25	25
CFG scale	7.5	4.5
Activation window K	4	4
Ramp coefficients ρ	$[0.2, 0.4, 0.6, 0.8]$	$[0.25, 0.5, 0.75, 1.0]$
Hardware	NVIDIA RTX A6000	NVIDIA RTX A6000

B First-Order Propagation Analysis

This section gives the derivation underlying the local sensitivity argument in Section 3.3. Consider a differentiable one-step reverse solver of the form

$$z_{j+1} = \mathcal{S}_j(z_j, \widehat{\epsilon}_j). \tag{8}$$

Let $\bar{z}_j$ denote the LoRA-off trajectory, z_j the watermark-enabled trajectory, and $d_j = z_j - \bar{z}_j$ their difference. Let $\widehat{\epsilon}_j^{\text{base}}(z)$ and $\widehat{\epsilon}_j^{\text{wm}}(z)$ denote the denoiser predictions with the LoRA disabled and enabled, respectively, including classifier-free guidance when it is used. We define the direct prediction perturbation by evaluating both models at the same watermark-enabled state z_j:

$$u_j = \widehat{\epsilon}_j^{\text{wm}}(z_j) - \widehat{\epsilon}_j^{\text{base}}(z_j). \tag{9}$$

Define

$$H_j(z, v) = \mathcal{S}_j\big(z, \widehat{\epsilon}_j^{\text{base}}(z) + v\big)\,.$$

Then $z_{j+1} = H_j(z_j, u_j)$ and $\bar{z}_{j+1} = H_j(\bar{z}_j, \mathbf{0})$. The composite frozen-base transition is $F_j^0(z) = H_j(z, \mathbf{0})$. Let $J_j = DF_j^0(\bar{z}_j)$ denote its total Jacobian with respect to the latent state. This definition includes the dependence of the frozen denoiser prediction on its input. We further let $G_j = D_v H_j(\bar{z}_j, \mathbf{0})$ denote the Jacobian with respect to the prediction perturbation. Assuming H_j is twice continuously differentiable in a neighborhood of $(\bar{z}_j, \mathbf{0})$, a first-order expansion gives

$$d_{j+1} = J_j d_j + G_j u_j + \mathcal{O}\big(\|(d_j, u_j)\|^2\big)\,. \tag{10}$$

With $d_1 = \mathbf{0}$, recursively applying Equation 10 and retaining first-order terms yields

$$d_{N+1} \approx \sum_{i=1}^{N} \left(J_N J_{N-1} \cdots J_{i+1}\right) G_i u_i, \tag{11}$$

where the empty product for $i = N$ is the identity. Here, the u_i are evaluated along the watermark-enabled trajectory, whereas J_j and G_j are evaluated along the base trajectory. Hence, a perturbation introduced at step i is transported through $N - i$ subsequent solver transitions. Setting $q_i = \mathbf{0}_r$ over the first $N - K$ steps makes $u_i = \mathbf{0}$ over that prefix, leaving only the final K contributions in Equation 11.

Table 6: Distortion and AI-regeneration configurations.

Transformation	Category	Setting
Color jitter	Signal processing	Brightness/contrast/saturation $\in (0.9, 1.1)$, hue $\in (-0.1, 0.1)$
Gaussian blur	Signal processing	5×5 kernel, $\sigma = 1$
Additive noise	Signal processing	Gaussian, $\sigma = 0.05$ (in $[0, 1]$ scale)
JPEG compression	Signal processing	Quality 50
Random crop	Signal processing	Approximately 80% area, then resized
BM3D	Signal processing	$\sigma = 0.1$
Resize	Signal processing	$512 \to 256 \to 512$ (bilinear)
SDEdit	AI regeneration	Strength 0.1
Rinse-2	AI regeneration	Two successive SDEdit transformations
VAE-B	AI regeneration	`bmshj2018_hyperprior`, quality 3

This is a local first-order sensitivity analysis rather than a guarantee that every early perturbation has a larger final norm. Subsequent solver transitions can contract or cancel individual contributions. For multistep solvers, the same expansion applies to an augmented state containing the required history, including cached denoiser predictions; the latent deviation is obtained by projecting the augmented-state deviation. For stochastic solvers, the identity of the LoRA-off and watermark-enabled prefix trajectories assumes coupled random draws.

C Evaluation Protocols and Method Configurations

C.1 Detection and Paired-Fidelity Metrics

For every method, we generate 1,000 images from 100 prompts selected from PartiPrompts and 10 initial-noise seeds per prompt. Bit accuracy is computed as the fraction of payload bits correctly recovered over the generated evaluation set.

Following the released AquaLoRA evaluation protocol (Feng et al., 2024), we use payload-length-specific binomial thresholds corresponding to a nominal false-positive rate (FPR) on the order of 10^{-6}. TPR is the fraction of generated images satisfying the corresponding detection criterion.

Paired fidelity is computed between watermark-enabled outputs and their matched watermark-free outputs generated with the same prompt, initial-noise seed, solver, and sampling configuration. We report PSNR, SSIM, and LPIPS. For distributional quality, we compute

$$\Delta\mathrm{FID} = \mathrm{FID}_{\mathrm{wm}} - \mathrm{FID}_{\mathrm{base}}. \tag{12}$$

We use matched 5,000-image watermarked and base-generated sets produced from COCO2017 validation captions. The corresponding 5,000 COCO2017 validation images serve as the real-image reference. We interpret $\Delta\mathrm{FID}$ only within a backbone. Small negative values are not treated as conclusive quality improvements without uncertainty estimates.

C.2 Distortion and AI-Regeneration Settings

Table 6 summarizes the transformations and evaluation settings applied before message decoding. We use BM3D (Dabov et al., 2007) and follow the AI-regeneration protocols of WAVES (An et al., 2024). SDEdit follows the diffusion-based editing procedure of Meng et al. (2022). Rinse-2 applies two successive SDEdit transformations, following the repeated-regeneration setting of WAVES. VAE-B follows the VAE-regeneration setting of WAVES and uses the bmshj2018 scale-hyperprior model (Ballé et al., 2018).

C.3 Method-Specific Evaluation Configurations

For every method, the watermarked and watermark-free outputs use the same prompt, initial-noise seed, solver, and sampling configuration. Table 7 summarizes the method-specific backbone and sampling settings. The evaluated configurations use PNDM (Liu et al., 2022), DDIM (Song et al., 2021), or DPM-Solver++ (2M) (Lu et al., 2025), following the corresponding released implementations. ResMark training and sampling configurations are reported separately in Table 5.

Table 7: Method-specific evaluation configurations. Stable Signature, LaWa, MOLM, and AquaLoRA are reproduced by fine-tuning SD1.5 using released code. StableGuard uses its officially released SD2.1-base checkpoint.

Method	Backbone	Resolution	Solver	Steps	CFG scale
Stable Signature	SD1.5	512×512	PNDM	50	7.5
LaWa	SD1.5	512×512	DDIM	50	7.5
MOLM	SD1.5	512×512	PNDM	50	7.5
AquaLoRA	SD1.5	512×512	DPM-Solver++ (2M)	25	7.5
StableGuard	SD2.1-base	512×512	PNDM	50	7.5
ResMark (Ours)	SD1.5	512×512	DPM-Solver++ (2M)	25	7.5
ResMark (Ours)	PixArt-α	512×512	DPM-Solver++ (2M)	25	4.5

Table 8: Effect of LoRA rank on PixArt under LNT and TRI. All configurations are evaluated with $K = 4$ late soft activation. The shaded row denotes the selected configuration.

Rank	Detection		Paired fidelity		
	Bit-Acc.↑	TPR↑	PSNR↑	SSIM↑	LPIPS↓
16	64.25	0.001	32.73	0.967	0.0205
32	74.66	0.069	30.27	0.952	0.0334
64	97.77	0.997	32.24	0.964	0.0255
128	98.39	1.000	32.69	0.967	0.0232
256	98.33	0.998	32.34	0.963	0.0255

D Additional Ablations and Diagnostics

D.1 LoRA Rank Ablation

Table 8 reports the complete LoRA rank sweep on PixArt. All configurations are trained with LNT and TRI and evaluated using the same $K = 4$ late soft activation schedule, with only the rank varied. The results confirm that low ranks provide insufficient capacity for reliable watermark insertion, even when some paired-fidelity metrics appear favorable. Rank 128 achieves the strongest detection while retaining high paired fidelity, whereas increasing the rank to 256 provides no further benefit. We therefore use rank 128 for the PixArt configuration.

D.2 Late-Step Activation Window Ablation

Table 9 evaluates the inference activation window using the same PixArt LNT–TRI checkpoint. All configurations use a monotone soft ramp over their respective active windows. Increasing K activates the watermark LoRA at earlier solver steps and allows its perturbations to propagate through more subsequent transitions.

Because the LNT interval remains fixed to $t \in \{0, \ldots, 160\}$, larger activation windows may also extend the LoRA beyond the low-noise regime encountered during training. The results therefore reflect both a longer perturbation-propagation path and an increased mismatch with the LNT training regime. Among the evaluated configurations, $K = 4$ provides the most favorable detection–fidelity balance.

D.3 Direct Final-Latent Addition Diagnostic

We compare ResMark with direct final-latent addition on PixArt. Given a watermark-free final latent, this diagnostic adds $\Delta(m)$ immediately before VAE decoding and does not use the watermark LoRA or any further denoising. It tests whether denoiser-side insertion provides advantages over directly decoding $z_0 + \Delta(m)$.

Table 10 reports performance on clean generated images. Under matched prompt–seed–payload tuples and the same watermark decoder, ResMark achieves higher bit accuracy than direct final-latent

Table 9: Effect of the late-step activation window on PixArt. All rows use the same adopted checkpoint trained with LNT and TRI on $t \in \{0, \ldots, 160\}$ and differ only in the activation window K, with the soft ramp defined by $\rho_i = i/K$ for $i = 1, \ldots, K$.

Schedule	Bit-Acc. (%)↑	TPR	PSNR	SSIM	LPIPS↓
$K = 4$, soft ramp	98.39	**1.000**	**32.69**	**0.967**	**0.0232**
$K = 8$, soft ramp	98.89	0.999	30.21	0.948	0.0399
$K = 12$, soft ramp	**99.07**	0.999	27.82	0.918	0.0630

Table 10: Direct final-latent addition diagnostic on clean generated images. Both rows use the same 1,000 prompt–seed–payload tuples and watermark decoder. Paired fidelity is measured against the matched LoRA-off generations.

Method	**Bit-Acc. (%)**	**TPR**	**PSNR**	**SSIM**	**LPIPS**
Direct $z_0 + \Delta(m)$	96.92	0.997	31.89	0.952	0.0250
ResMark (PixArt)	**98.39**	**1.000**	**32.69**	**0.967**	**0.0232**

Table 11: Robustness of the direct final-latent addition diagnostic. Both methods use the same 1,000 prompt–seed–payload tuples, distortion parameters, and watermark decoder.

Method	**Metric**	**Signal processing**							**AI regeneration**		
		Color jitter	Blur	Noise	JPEG	Crop	BM3D	Resize	SDEdit	Rinse-2	VAE-B
Direct $z_0 + \Delta(m)$	Bit-Acc. (%)	93.55	96.86	95.12	95.25	92.77	96.16	96.68	83.56	72.25	93.48
ResMark (PixArt)	Bit-Acc. (%)	**96.18**	**98.38**	**97.42**	**97.44**	**95.64**	**97.96**	**98.28**	**90.65**	**81.84**	**96.38**

addition (98.39% vs. 96.92%) and a slightly higher TPR (1.000 vs. 0.997). It also provides better paired fidelity, with higher PSNR and SSIM and lower LPIPS.

Table 11 compares the two approaches under matched distortion settings. ResMark achieves higher bit accuracy under all ten transformations, with particularly pronounced improvements under SDEdit and Rinse-2. Together, these results show that denoiser-side residual insertion improves both clean-image performance and post-transformation robustness over direct final-latent addition at the evaluated operating points.

E Additional Qualitative Results

We provide additional matched-seed qualitative comparisons on SD1.5 and PixArt-α.

E.1 Qualitative Results on SD1.5

Figure 3 provides additional comparisons on SD1.5. Each column uses the same prompt, initial-noise seed, solver, and sampling configuration across the three rows.

E.2 Qualitative Results on PixArt-α

Figure 4 provides additional comparisons on PixArt-α. Each column uses the same prompt, initial-noise seed, solver, and sampling configuration across the three rows.

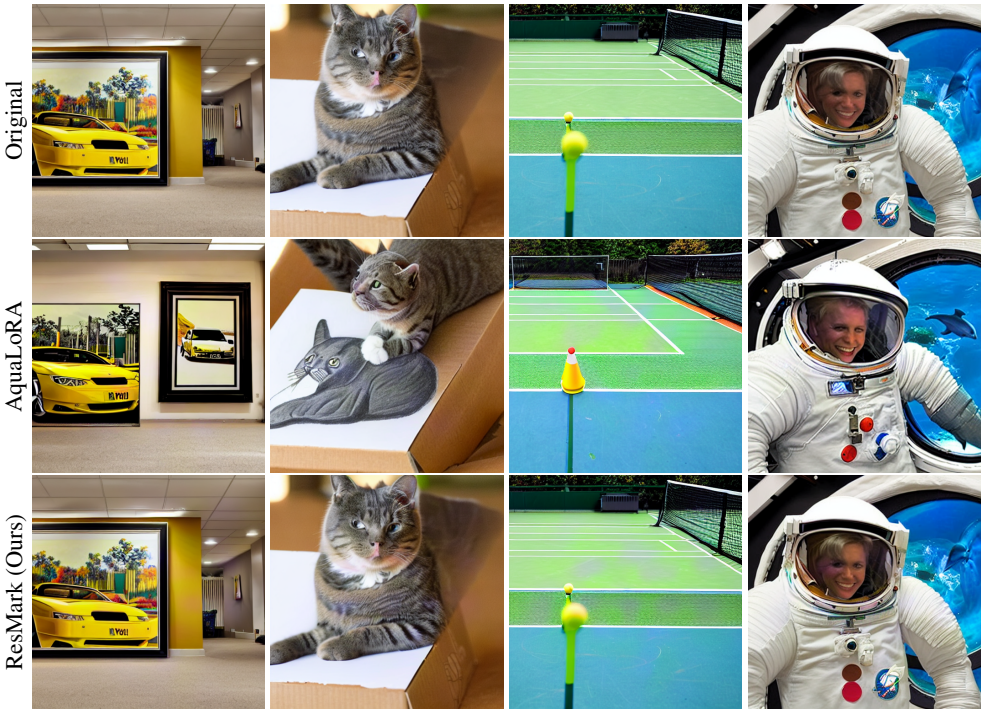

Figure 3: **Additional controlled qualitative comparisons on SD1.5.** Each column uses an identical prompt, initial-noise seed, solver, and sampling configuration. AquaLoRA uses full-trajectory activation, whereas ResMark uses LNT, TRI, and late soft activation.

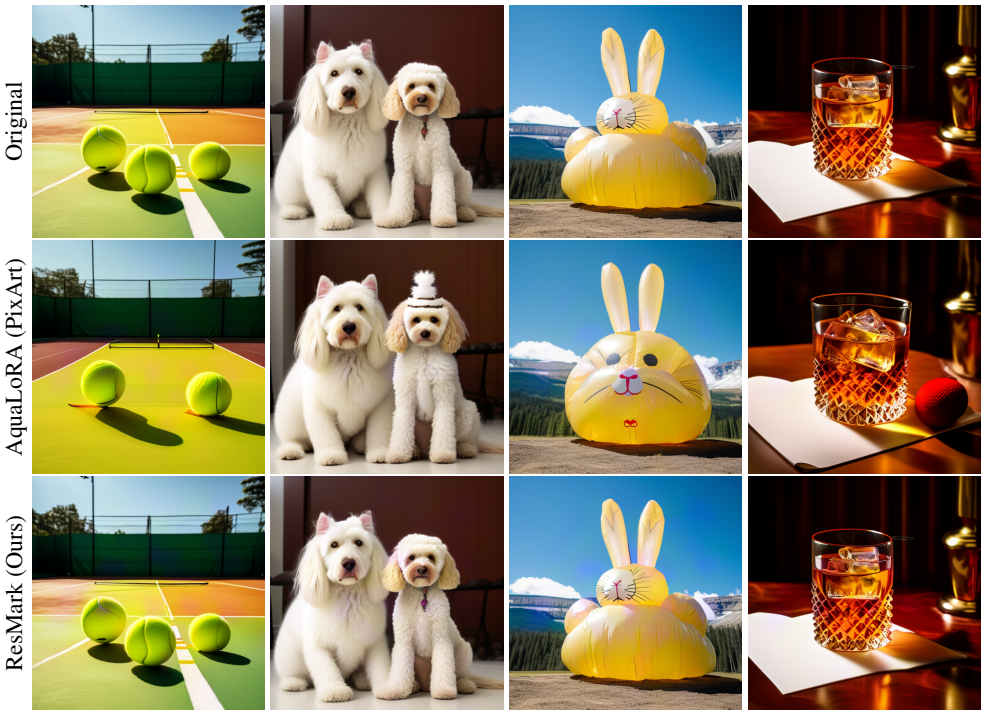

Figure 4: **Additional controlled qualitative comparisons on PixArt-α.** Each column uses an identical prompt, initial-noise seed, solver, and sampling configuration. The middle row shows our PixArt implementation of AquaLoRA using full-timestep PPFT and full-trajectory activation, whereas ResMark uses LNT, TRI, and late soft activation.