ZEROSTEP: RETHINKING DIFFUSION INVERSION FOR WATERMARK VERIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Embedding watermarks into the initial noise provides a natural mechanism for copyright protection and source attribution in diffusion generation. Yet existing verification methods still invert the generated image back to the initial watermarked noise, making costly diffusion inversion a standard step in watermark recovery. We revisit this requirement by tracing watermark evidence along the generation trajectory and find that, although similarity to the initial watermarked noise drops rapidly, watermark recoverability remains high even near the generation endpoint. This observation motivates ZeroStep, a zero-step watermark verification method that removes diffusion inversion entirely. Our method trains a lightweight LoRA adapter on the VAE encoder with trajectory supervision across timesteps and latent components, enabling direct watermark extraction without diffusion inversion. For robustness during online image distribution, we further leverage paired clean and distorted views to guide distortion-robust training. Experiments across diffusion and flow-matching backbones show that our method consistently outperforms state-of-the-art inversion-based methods. Notably, on the challenging SD v3.5 backbone, it improves watermark bit accuracy from 95.88% to 99.11% while reducing the deployed model size from 2.5B to 35.01M parameters. Moreover, the learned adapter can be reused across compatible model versions and scales, demonstrating strong cross-model adaptability. Code is available at `https://anonymous.4open.science/r/ZeroStep-Release`.

1 INTRODUCTION

Diffusion models have made high-quality visual content much easier to generate, while also increasing the need for provenance authentication, copyright attribution, and accountability for misuse (Ho et al., 2020; Fernandez et al., 2023; Podell et al., 2024). *Initial noise watermarking* addresses these needs by embedding watermark information into the initial Gaussian latent, after which sampling proceeds normally with the original generator (Wen et al., 2023; Ci et al., 2024; Yang et al., 2024). Recent work has extended this paradigm beyond binary detection to support multi-key and message-level attribution, as well as distribution-preserving and semantic-aware watermarking (Gunn et al., 2025; Arabi et al., 2025a;b). As these watermarked images circulate widely across the Internet, provenance tracing must process large volumes of verification queries efficiently and reliably, even when the content has undergone diverse transformations.

At verification time, although their embedding mechanisms differ, existing *initial noise watermarking* methods largely share the same inversion-based pipeline. Given a query image, the variational autoencoder (VAE) first maps it to the latent space; multi-step DDIM inversion (Song et al., 2021) then reconstructs the initial noise with repeated denoiser evaluations; and the verifier detects the watermark from this estimate (Wen et al., 2023; Yang et al., 2024; Arabi et al., 2025b; Hu et al., 2026). This process is costly at scale. Existing efforts to accelerate inversion, however, have largely focused on image reconstruction and editing, where inversion fidelity is improved through iterative refinement, numerical optimization, or explicitly invertible solvers (Mokady et al., 2023; Wallace et al., 2023; Hong et al., 2024; Garibi et al., 2024; Wang et al., 2024), rather than on recovering watermark-bearing initial noise. More recently, FARI and OSI reduce this cost by collapsing inversion to a single model evaluation, substantially lowering the number of function evaluations (NFE) (Yang et al., 2026; Chen et al., 2026). However, they still keep inversion in the verification loop and invoke the diffusion denoiser to recover the watermark indirectly, without directly examining how watermark recoverability evolves along the generation trajectory.

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.

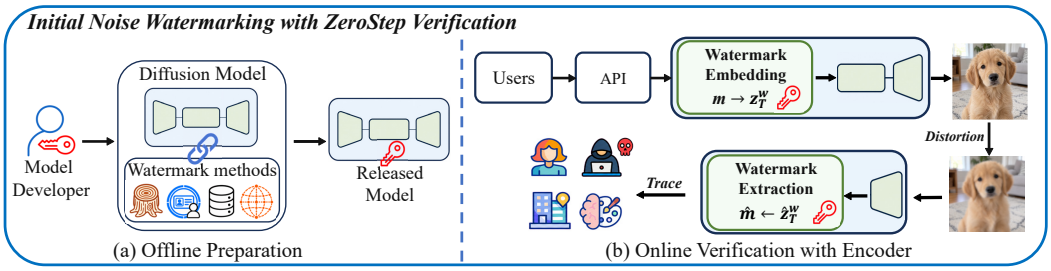


Figure 1: Overall workflow of initial noise watermarking with ZeroStep verification. (a) Offline preparation. (b) Online watermark embedding and direct verification with a single encoder pass.

To directly characterize how watermark evidence evolves, we track it along both the generation and inversion trajectories. We observe a clear mismatch between similarity to the initial watermarked noise and watermark recoverability. As denoising proceeds, intermediate latents rapidly lose similarity to the initial watermarked noise, while verification performance degrades much more slowly. This gap persists through the image endpoint and into early inversion, where the VAE latent and early inversion states retain sufficient watermark evidence despite low similarity to the initial noise. These results show that watermark evidence is not confined to the initial noise but persists across a broad range of latent states. Since the VAE encoder naturally operates at this latent endpoint, its output occupies the same latent space as the states traversed during subsequent inversion, implicitly providing a structured projection for watermark verification.

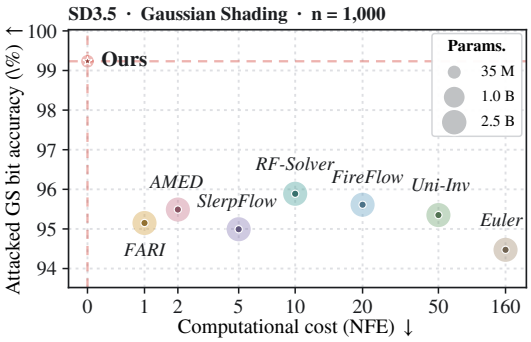


Figure 2: Comparison of ZeroStep (SD3.5) with one-step, few-step, and multi-step inversion methods. Results are reported over $n = 1{,}000$ samples in terms of Gaussian Shading bit accuracy and deployment parameter count.

Building on these observations, we propose Zero-Step Watermark Verification (ZeroStep), as illustrated in Figure 1, which removes diffusion inversion from the verification pipeline. Specifically, we introduce Watermark Trajectory Distillation, which uses intermediate latents across timesteps as supervision to help the VAE encoder capture watermark evidence that persists along the generation trajectory. To improve robustness, we further introduce distortion-robust training with diverse image transformations, enabling reliable watermark recovery from degraded inputs. During training, we update only lightweight LoRA parameters in the VAE encoder, allowing it to directly produce latent representations compatible with the original watermark verifier. At verification time, a query image requires only a single pass through the adapted VAE encoder before being processed by the original detector or decoder. This lightweight design translates into substantial efficiency gains, as shown in Figure 2. On the more challenging SD v3.5 setting, ZeroStep is up to $4.91\times$ faster than the fastest prior method and reduces the deployed model size from 2.5B to 35.01M parameters, while achieving 99.11% watermark verification accuracy and outperforming the strongest baseline by 3.23 percentage points. Interestingly, we further find that when model variants share a compatible latent representation space, the same VAE-LoRA adapter can be reused across versions and scales.

In summary, our main contributions are as follows:

- We reveal that watermark recoverability persists along the generation trajectory even as similarity to the initial watermarked noise rapidly diminishes, showing that the endpoint latent still retains sufficient watermark evidence for verification.
- We propose ZeroStep, an inversion-free watermark verification framework that adapts the VAE encoder with lightweight LoRA through Watermark Trajectory Distillation and distortion-robust training, enabling direct verification from the endpoint latent.
- Extensive experiments across diffusion and flow-matching backbones and diverse watermarking approaches show that our method substantially improves verification performance with only 0.738M trainable parameters. The learned adapter can also be reused across compatible model versions and scales.

2 Related Works

Diffusion Generation and Inversion. Diffusion models generate images by progressively denoising Gaussian noise (Ho et al., 2020; Song et al., 2021), while latent diffusion carries this process into the latent space of a pretrained VAE (Rombach et al., 2022). Subsequent work reduces the number of function evaluations (NFE) required for sampling. DPM-Solver (Lu et al., 2022) accelerates generation with high-order ODE solvers, UniPC (Zhao et al., 2023) uses predictor and corrector updates, and AMED (Zhou et al., 2024) improves sampling under extremely low NFE. In parallel, inversion maps generated images back toward earlier latent or noise states. EDICT (Wallace et al., 2023), BELM (Wang et al., 2024), and ExactDPM (Hong et al., 2024) improve inversion accuracy through enhanced trajectory reversibility or reduced numerical error. More recently, flow matching and rectified flow formulate generation as integration along a learned vector field (Lipman et al., 2023; Liu et al., 2023) and underpin models such as SD3 and FLUX (Esser et al., 2024; Labs et al., 2025). Corresponding inversion methods, including RF-Solver (Wang et al., 2025), FireFlow (Deng et al., 2025), and Uni-Inv (Jiao et al., 2026), likewise seek more accurate or efficient trajectory reversal. Across both paradigms, inversion ultimately traces generated outputs back toward their earlier latent states. As generated content proliferates, this traceability becomes increasingly important for provenance and accountability.

Watermarking for Diffusion Models. Watermarking provides a direct means of attaching verifiable provenance information to generated content. *Initial noise watermarking* directly encodes watermark information into the Gaussian noise used to initialize sampling while leaving the subsequent generation process unchanged. Tree-Ring (Wen et al., 2023) established this paradigm by embedding structured Fourier patterns into the initial noise. Subsequent work further extends this paradigm (Chen et al., 2025; Li et al., 2025), such as RingID (Ci et al., 2024) for multiple keys, WIND (Arabi et al., 2025a) for larger key spaces, and SEAL (Arabi et al., 2025b) for verification tied to image semantics. As watermarking applications move from presence detection toward finer-grained provenance tracing and more robust deployment (Yang et al., 2025; Li et al., 2025), Gaussian Shading (Yang et al., 2024) enables tracing of multiple bits by using repetition codes and a stream cipher to sample watermarked latents from a standard Gaussian distribution. However, using distinct keys and nonces for each image introduces substantial management overhead. PRC watermarking (Gunn et al., 2025) subsequently replaces the stream cipher with pseudorandom codes designed for error correction and generated from a fixed secret key, reducing this overhead. Spherical Watermark (Hu et al., 2026) further removes encryption and separate key storage for each image while preserving the Gaussian distribution of the watermarked noise. Despite different embedding mechanisms, verification still relies on inversion to recover initial noise, making denoiser evaluations a major computational bottleneck at scale. FARI (Yang et al., 2026) and OSI (Chen et al., 2026) reduce this process to a single model evaluation, substantially lowering verification cost, but still retain inversion as an intermediate step. In contrast, our method removes inversion entirely and directly extracts watermark information from the VAE latent.

3 Motivation

Watermark Generation. *Initial noise watermarking* places the watermark in the starting latent rather than modifying the generated image afterward. Given a watermark message m and a secret key K, the watermark encoder $\mathcal{W}_K$ produces a watermarked initial latent z_T^w. Generation then follows the standard diffusion process, starting from z_T^w and progressively denoising the latent from timestep T to 0, after which the resulting z_0^w is decoded by the VAE into a watermarked image x^w:

$$z_T^w = \mathcal{W}_K(m), \qquad z_T^w \to \cdots \to z_0^w, \qquad x^w = D_{\mathrm{VAE}}(z_0^w) \tag{1}$$

The watermark embedded in the initial noise therefore propagates through the full denoising trajectory before reaching the generated image.

Inversion-Based Verification. At verification time, existing methods typically recover an estimate of the latent in which the watermark was originally embedded. Given a query image $\hat{x}$, the VAE encoder first produces its latent representation $\hat{z}_0^w$. DDIM inversion then propagates $\hat{z}_0^w$ back toward the initial noise, yielding an estimate $\hat{z}_T^w$, from which the watermark verifier $\mathcal{R}_K$ recovers $\hat{m}$:

$$\hat{z}_0^w = E_{\mathrm{VAE}}(\hat{x}), \qquad \hat{z}_0^w \to \cdots \to \hat{z}_T^w, \qquad \hat{m} = \mathcal{R}_K(\hat{z}_T^w) \tag{2}$$

Inversion therefore maps the query image back to a latent representation for watermark verification, but how the watermark evidence evolves across the intermediate states remains unclear.

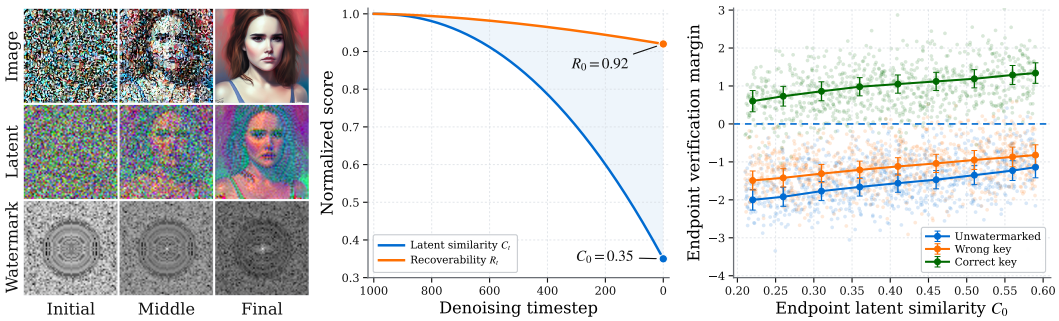


Figure 3: Watermark recoverability persists despite substantial latent drift during generation. **Left**: image, latent, and watermark representations along the denoising trajectory. **Middle**: latent similarity C_t drops rapidly, whereas Tree-Ring recoverability R_t remains high toward the image endpoint. **Right**: endpoint latents remain strongly verifiable even at low similarity to the initial watermarked noise.

Trajectory Visualization. To visualize this evolution, we inspect representative states along the generation trajectory. As shown in Fig. 3(left), each timestep is visualized through its latent state, decoded image, and corresponding Tree-Ring pattern. As denoising proceeds, image structure becomes progressively clearer in the latent and the decoded content becomes recognizable. In contrast, the Tree-Ring pattern decays more slowly and remains discernible even after substantial image structure has formed. This contrast suggests that latent evolution and watermark retention are not synchronized, motivating a comparison between latent similarity and watermark recoverability.

Similarity vs. Recoverability. To quantify this mismatch, we measure latent similarity and watermark recoverability at each timestep. We first use cosine similarity to measure the global directional alignment between an intermediate watermarked latent z_t^w and the initial watermarked latent z_T^w:

$$C_t = \frac{\langle z_t^w, z_T^w \rangle}{\|z_t^w\|_2 \|z_T^w\|_2} \tag{3}$$

We then apply the original watermark verifier directly to z_t^w. For Tree-Ring, the verifier computes a detection distance δ for each latent, with smaller values indicating stronger watermark agreement. Given a target false positive rate α, we calibrate the threshold τ_t^α on null samples and define recoverability as the fraction of watermarked samples below this threshold:

$$R_t^\alpha = \frac{1}{N_w} \sum_{i=1}^{N_w} \mathbb{I}\left(\delta_{t,i}^w \le \tau_t^\alpha\right) \tag{4}$$

As shown in Fig. 3(middle), C_t drops rapidly during denoising, whereas R_t remains high over a much longer portion of the trajectory. This divergence shows that watermark recoverability can persist even after similarity to z_T^w has substantially diminished.

Endpoint Analysis. This gap persists at the endpoint z_0^w, where similarity to z_T^w is low but the watermark remains verifiable. We quantify this endpoint evidence using a standardized margin:

$$M_0^K = \frac{\tau_0^\alpha - \delta_0^K}{\sigma_0^{\text{null}}} \tag{5}$$

where δ_0^K is the Tree-Ring detection distance under key K, τ_0^α is the threshold at false positive rate α, and σ_0^{null} is the standard deviation of null distances. A positive M_0^K indicates successful verification. As shown in Fig. 3(right), correct keys retain positive margins even at low C_0, whereas wrong keys and unwatermarked samples yield substantially lower margins. This confirms that the endpoint latent retains discriminative watermark structure despite low similarity to the initial watermarked noise, making it a promising source for direct watermark verification.

4 METHOD

Overview. Motivated by the above observations, we propose **Zero-Step Watermark Verification (ZeroStep)**, as illustrated in Fig.4. ZeroStep trains the VAE encoder to directly recover a latent for watermark verification from a query image. During training, intermediate latents with high watermark retention from the frozen diffusion trajectory serve as supervision, while common image distortions are applied to improve robustness.

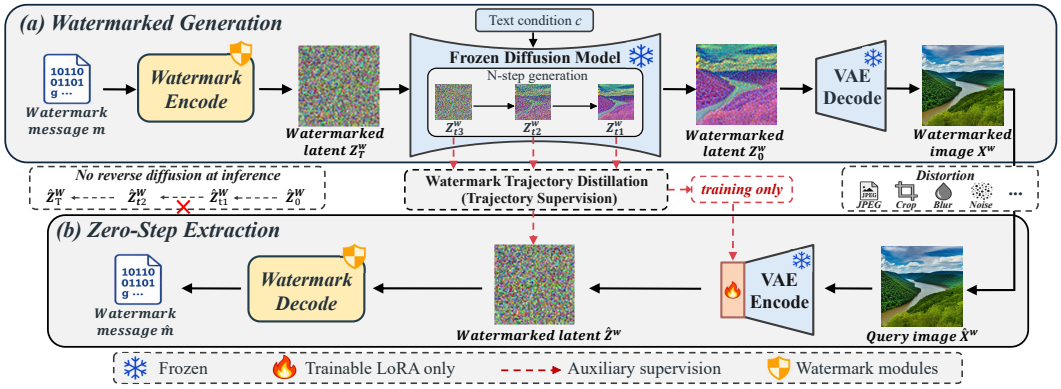


Figure 4: Overview of Zero-Step Watermark Verification (ZeroStep). (a) Watermarked generation remains unchanged, while intermediate diffusion states provide retention-aware supervision only during training. (b) Adapts only the VAE encoder with LoRA and distortion-robust training, allowing a query image to be mapped directly to a latent for the watermark decoder without reverse diffusion.

Watermark Trajectory Distillation. Our analysis in Section 3 reveals a broad region of the generation trajectory where the latent has already deviated substantially from the initial watermarked noise z_T^w, while watermark recoverability remains high. This suggests that supervision need not be restricted to a single initial state. Instead, we select teacher states from the trajectory according to both their verification reliability and their deviation from z_T^w. Specifically, for each candidate timestep $t \in \mathcal{T}$, we define

$$w_t = R_t(1 - C_t), \qquad p_t = \frac{w_t}{\sum_{s \in \mathcal{T}} w_s} \tag{6}$$

where R_t denotes watermark recoverability and C_t denotes the similarity between z_t^w and z_T^w. The recoverability term favors states that preserve reliable watermark evidence, while $1-C_t$ gives greater weight to states that have moved farther from the initial noise. The normalized distribution p_t therefore concentrates supervision on trajectory states that remain verifiable despite substantial latent change. We sample a teacher timestep $t \sim p_t$ and regress the clean encoder prediction toward the corresponding state z_t^w:

$$\mathcal{L}_{\text{traj}} = \mathbb{E}_{t \sim p_t}\left[\frac{1}{d}\left\|E_\phi(x^w) - z_t^w\right\|_2^2\right] \tag{7}$$

where d denotes the latent dimensionality. This objective supervises the encoder with trajectory states that retain strong watermark evidence, rather than only the initial watermarked latent.

Distortion-Robust Training. Query images may undergo compression, cropping, blurring, noise, or other transformations before verification, which can shift the encoder output away from a reliable watermark representation. We therefore construct clean and distorted views of the same watermarked image and jointly constrain the distorted prediction through representation consistency and watermark alignment. Given x^w, we denote the clean prediction by $\hat{z} = E_\phi(x^w)$ and the distorted prediction by $\hat{z}^{\mathcal{A}} = E_\phi(\mathcal{A}(x^w))$, where $\mathcal{A} \sim \mathcal{P}_{\mathcal{A}}$ is sampled from the distortion distribution. The objective is:

$$\mathcal{L}_{\text{Distort}} = \mathbb{E}_{\mathcal{A} \sim \mathcal{P}_{\mathcal{A}}}\left[\underbrace{\frac{1}{d}\left\|\hat{z}^{\mathcal{A}} - \hat{z}\right\|_2^2}_{\text{representation consistency}} + \lambda_{\text{w}} \underbrace{\frac{1}{d}\left\|\hat{z}^{\mathcal{A}} - z_T^w\right\|_2^2}_{\text{watermark alignment}}\right] \tag{8}$$

The consistency term limits latent drift between clean and distorted observations. However, consistency alone does not ensure that the shared representation remains informative for watermark verification. The alignment term therefore anchors the distorted prediction to the original watermarked latent, preserving watermark evidence under input degradation. In practice, $\mathcal{P}_{\mathcal{A}}$ covers common transformations including JPEG compression, crop, resize, blur, noise, and brightness changes.

Parameter-Efficient Adaptation. ZeroStep adapts the VAE encoder with LoRA (Hu et al., 2022), while keeping the pretrained VAE weights, diffusion model, and watermark verifier frozen. For a

pretrained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, LoRA introduces a low-rank update:

$$W = W_0 + BA \tag{9}$$

where $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$, and $r \ll \min(d, k)$. Only A and B are optimized during training.

We further concentrate supervision on latent components that carry stronger watermark evidence. Beyond selecting informative timesteps, watermark evidence is also unevenly distributed across latent components. As shown in Fig. 5, components with larger predicted magnitudes retain more correct watermark evidence.

$$\bar{w}_j = \frac{|\hat{z}_j|^\gamma}{\sum_{k=1}^{d} |\hat{z}_k|^\gamma} \tag{10}$$

where γ controls the weighting strength and $\gamma = 0$ recovers uniform weighting.

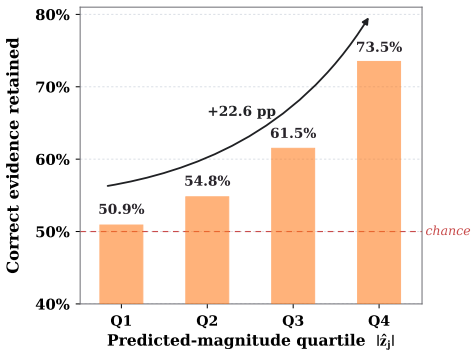


Figure 5: Watermark evidence across quartiles of predicted latent magnitude, showing that higher-magnitude components retain a larger share of correct watermark evidence.

Finally, we combine trajectory supervision and distortion robustness under the weight:

$$\mathcal{L}_{\text{total}} = \mathcal{L}^{\bar{w}}_{\text{traj}} + \lambda_{\text{D}} \mathcal{L}^{\bar{w}}_{\text{Distort}} \tag{11}$$

where $\bar{w}$ indicates that the corresponding latent errors are weighted by Eq. 10. Further implementation details and hyperparameter analyses are provided in Section 5.1 and Appendix A.

5 Experiments

5.1 Experimental Setup

Diffusion Backbones.[1] We conduct experiments on SD v1.5 (Rombach et al., 2022), SD v2.1 (Rombach et al., 2022), SDXL-Turbo (Sauer et al., 2024), and SD v3.5 (Esser et al., 2024), covering both diffusion and flow matching generators. Training follows the native generation trajectory of each backbone, using denoising trajectories for the first three models and the flow trajectory for SD v3.5. The generative models remain frozen, with updates restricted to the LoRA parameters in the VAE encoder. Additional backbones and watermarking schemes are reported in Appendix B.

Watermarking Schemes. We use Tree-Ring (Wen et al., 2023) and Gaussian Shading (Yang et al., 2024) to evaluate structured watermark detection and multi-bit recovery, respectively. Their original embedding and verification procedures remain unchanged, and we modify only how the verification latent is obtained, isolating the effect of watermark extraction.

Training and Evaluation. We train for 1,000 steps on 1,000 prompts from MS-COCO 2017 (Lin et al., 2014) and evaluate on a separate set of 1,000 prompts from Stable-Diffusion-Prompts (SDP)[2]. Adam is used with a learning rate of 10^{-5}, batch size of 4, and LoRA rank $r = 8$, with $\gamma = 1$ and $\lambda_{\text{D}} = 1$. Each training sample is randomly augmented with one of nine common image distortions at a shared moderate severity, with exact settings provided in the Appendix. All backbones generate 512×512 images and otherwise follow their default sampling configurations. Experiments are conducted on a single NVIDIA RTX 5090, with evaluation on both clean and distorted queries.

Baselines and Metrics. We compare against multistep inversion methods, including DDIM (Song et al., 2021), EDICT (Wallace et al., 2023), BELM (Wang et al., 2024), and ExactDPM (Hong et al., 2024); low-step numerical inversion based on direct step reduction or fast solvers, such as AMED (Zhou et al., 2024); and learned few-step or one-step baselines, including LCM-LoRA (Luo et al., 2023), DMD2 (Yin et al., 2024), and FARI (Yang et al., 2026). For SD v3.5, we additionally include flow inversion methods such as Uni-Inv (Jiao et al., 2026), FireFlow (Deng et al., 2025), RF-Solver (Wang et al., 2025), and SlerpFlow (Duan et al., 2026). Official implementations and recommended settings are used whenever available; otherwise, we reproduce the reported configurations. Tree-Ring is evaluated by TPR at an FPR of 10^{-3}, while Gaussian Shading is evaluated by bit accuracy. Efficiency is measured by verification latency, throughput, and denoiser NFE.

[1] https://huggingface.co/stabilityai

[2] https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts

Table 1: Comparison of methods on downstream watermarking methods under various image distortions.

DM	Methods	NFE	Clean	Avg.	Jpeg	R.Crop	R.Drop	Resize	G.Blur	M.Blur	G.Noise	S&P	Bright
Bit Accuracy of Gaussian Shading Watermark													
SD v1.5	DDIM	50	0.9999	0.9752	0.9889	0.9781	0.9736	0.9975	0.9873	0.9983	0.9609	0.9354	0.9567
		1	0.9999	0.9376	0.9703	0.8859	0.8808	0.9906	0.9585	0.9934	0.9398	0.9105	0.9085
	EDICT	50	**1.0000**	0.9637	0.9786	0.9656	0.9568	0.9969	0.9807	0.9985	0.9390	0.9124	0.9450
	BELM	50	0.9991	0.9465	0.9847	0.8960	0.8958	0.9939	0.9617	0.9956	0.9355	0.9275	0.9278
	AMED	2	**1.0000**	0.9656	0.9807	0.9528	0.9462	0.9970	0.9808	0.9989	0.9587	0.9346	0.9410
	LCM-LoRA	2	0.9999	0.9541	0.9819	0.9352	0.9308	0.9924	0.9668	0.9955	0.9311	0.9030	0.9504
	DMD2	1	0.9988	0.9209	0.9760	0.8446	0.8241	0.9792	0.9209	0.9836	0.9252	0.9007	0.9336
	FARI	1	1.0000	0.9834	0.9935	0.9777	0.9761	0.9990	0.9957	0.9992	0.9836	0.9649	0.9612
	ZeroStep	**0**	**1.0000**	**0.9989**	**0.9994**	**0.9998**	**0.9997**	**0.9997**	**0.9989**	**0.9998**	**0.9998**	**0.9999**	**0.9933**
SD v2.1	DDIM	50	**1.0000**	0.9728	0.9892	0.9752	0.9669	0.9980	0.9860	0.9991	0.9590	0.9373	0.9447
		1	0.9987	0.9290	0.9755	0.8841	0.8637	0.9748	0.9173	0.9819	0.9280	0.9069	0.9284
	EDICT	50	**1.0000**	0.9624	0.9773	0.9639	0.9558	0.9963	0.9805	0.9983	0.9396	0.9104	0.9395
	BELM	50	0.9990	0.9459	0.9847	0.8923	0.8938	0.9933	0.9602	0.9952	0.9333	0.9269	0.9334
	AMED	2	**1.0000**	0.9662	0.9813	0.9559	0.9495	0.9970	0.9805	0.9989	0.9585	0.9357	0.9384
	ExactDPM	160	**1.0000**	0.9681	0.9831	0.9675	0.9599	0.9974	0.9815	0.9987	0.9653	0.9241	0.9354
	FARI	1	1.0000	0.9824	0.9941	0.9771	0.9700	0.9992	0.9956	0.9994	0.9815	0.9659	0.9588
	ZeroStep	**0**	**1.0000**	**0.9983**	**0.9990**	**0.9997**	**0.9996**	**0.9999**	**0.9990**	**0.9999**	**0.9996**	**0.9999**	**0.9884**
SDXL-Turbo	DDIM	50	0.9900	0.8854	0.9402	0.8079	0.8282	0.9389	0.8549	0.9510	0.9479	0.8442	0.8555
		1	0.9755	0.8575	0.9142	0.7449	0.7552	0.9370	0.8841	0.9346	0.8935	0.8229	0.8315
	EDICT	50	0.9476	0.8183	0.8822	0.7197	0.7360	0.8591	0.8112	0.8803	0.8929	0.7799	0.8030
	BELM	50	0.8927	0.7910	0.8389	0.6767	0.6886	0.8779	0.8152	0.8658	0.8429	0.7443	0.7689
	FARI	1	0.9998	0.9511	0.9903	0.9115	0.9161	0.9871	**0.9370**	0.9922	0.9801	0.9354	0.9102
	ZeroStep	**0**	**1.0000**	**0.9791**	**0.9912**	**0.9838**	**0.9867**	**0.9906**	**0.9360**	**0.9944**	**0.9946**	**0.9740**	**0.9607**
SDV3.5	Euler	160	0.9999	0.9447	0.9522	0.9848	0.9684	0.9362	0.8394	0.9559	0.9851	0.9018	0.9787
	Uni-Inv	50	0.9999	0.9535	0.9573	0.9878	0.9725	0.9554	0.8711	0.9681	0.9855	0.9032	0.9804
	FireFlow	20	0.9999	0.9561	0.9591	0.9884	0.9731	0.9597	0.8841	0.9705	0.9856	0.9029	0.9811
	RF-Solver	10	0.9999	0.9588	0.9618	0.9847	0.9678	0.9661	0.9111	0.9778	0.9859	0.8921	0.9816
	SlerpFlow	5	0.9998	0.9499	0.9625	0.9630	0.9479	0.9622	0.8895	0.9748	0.9814	0.8917	0.9762
	AMED	2	0.9999	0.9549	0.9540	0.9859	0.9690	0.9630	0.8917	0.9751	0.9819	0.8931	0.9804
	FARI	1	0.9995	0.9515	0.9671	0.9713	0.9469	0.9756	0.9384	0.9852	0.8765	0.9191	0.9833
	ZeroStep	**0**	**1.0000**	**0.9911**	**0.9924**	**0.9995**	**0.9963**	**0.9926**	**0.9619**	**0.9941**	**0.9970**	**0.9905**	**0.9954**
TPR @ FPR $= 10^{-3}$ **of Tree-Ring Watermark**													
SD v1.5	DDIM	50	**1.000**	0.949	0.989	**1.000**	**1.000**	0.999	0.996	**1.000**	0.636	0.946	0.972
		1	**1.000**	0.863	0.905	0.602	0.649	**1.000**	0.994	**1.000**	0.891	0.990	0.737
	EDICT	50	**1.000**	0.942	0.975	0.998	**1.000**	0.999	0.992	0.997	0.605	0.954	0.962
	BELM	50	0.933	0.592	0.768	0.032	0.054	0.889	0.873	0.865	0.384	0.852	0.608
	AMED	2	**1.000**	0.909	0.939	0.947	0.936	0.999	0.995	0.999	0.618	0.912	0.835
	LCM-LoRA	2	**1.000**	0.875	0.914	0.996	0.991	0.999	0.987	0.999	0.331	0.812	0.850
	DMD2	1	**1.000**	0.760	0.709	0.116	0.473	0.996	0.971	0.995	0.913	0.985	0.678
	FARI	1	1.000	0.997	1.000	1.000	1.000	1.000	1.000	1.000	0.980	1.000	0.992
	ZeroStep	**0**	**1.000**	**0.998**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.989**	**1.000**	**0.997**
SD v2.1	DDIM	50	**1.000**	0.962	0.993	**1.000**	**1.000**	**1.000**	0.997	**1.000**	0.726	0.982	0.960
		1	**1.000**	0.896	0.896	0.709	0.845	**1.000**	0.993	0.999	0.903	0.991	0.729
	EDICT	50	**1.000**	0.946	0.980	0.984	0.985	0.997	0.990	0.998	0.681	0.969	0.933
	BELM	50	0.882	0.543	0.721	0.001	0.000	0.803	0.787	0.801	0.417	0.812	0.541
	AMED	2	**1.000**	0.926	0.966	0.957	0.983	**1.000**	0.998	**1.000**	0.656	0.983	0.795
	FARI	1	1.000	0.997	0.999	1.000	1.000	1.000	0.999	1.000	0.979	0.999	0.993
	ZeroStep	**0**	**1.000**	**0.999**	**1.000**	**1.000**	**1.000**	**1.000**	**0.999**	**1.000**	**0.989**	**0.999**	**1.000**

5.2 Verification without Inversion

Table 1 shows that our method maintains strong watermark verification across different backbones and image distortions, matching or outperforming inversion-based baselines under the same settings. Methods designed for accurate inversion, such as EDICT (Wallace et al., 2023), BELM (Wang et al., 2024), and ExactDPM (Hong et al., 2024), remain competitive on clean inputs, but their gains in trajectory reversibility and numerical accuracy do not consistently carry over under crop, drop, and noise. Generation-oriented methods such as LCM-LoRA (Luo et al., 2023) and DMD2 (Yin et al., 2024) also transfer poorly to low-step inversion. This behavior reflects the gap between reconstructing the initial noise and recovering the representation required for watermark verification. By predicting this representation directly from the query image, our method achieves the strongest overall verification performance across the evaluated settings.

A similar pattern appears on the flow-matching backbone SD v3.5. Most flow inversion methods already approach saturated verification on clean images, whereas their robustness diverges once the inputs are distorted. Methods such as RF-Solver (Wang et al., 2025), FireFlow (Deng et al., 2025), Uni-Inv (Jiao et al., 2026), and SlerpFlow (Duan et al., 2026) improve inversion mainly by refining the numerical solution of the reverse flow dynamics, whereas our training directly uses intermediate states from the native flow trajectory as supervision, allowing the encoder to learn verification representations without explicitly reconstructing the initial state. This training strategy therefore extends naturally to flow-matching backbones without relying on a specific inversion solver.

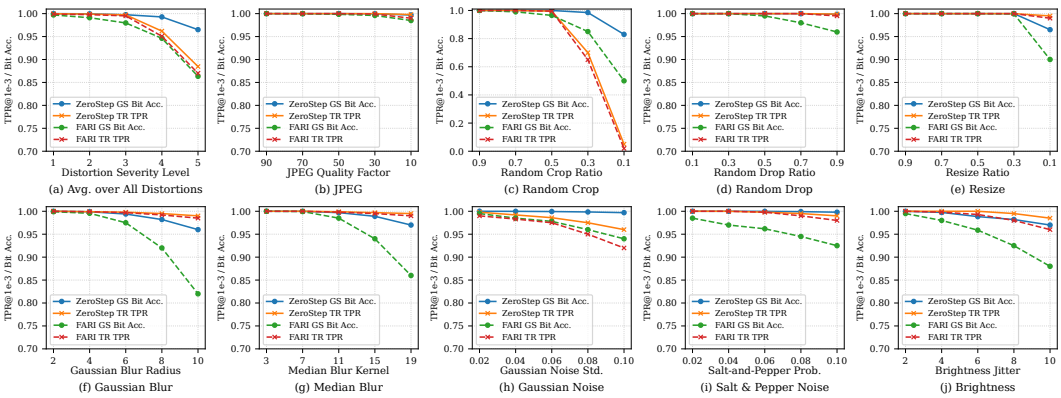


Figure 6: Robustness under increasing distortion severity on SD v2.1.

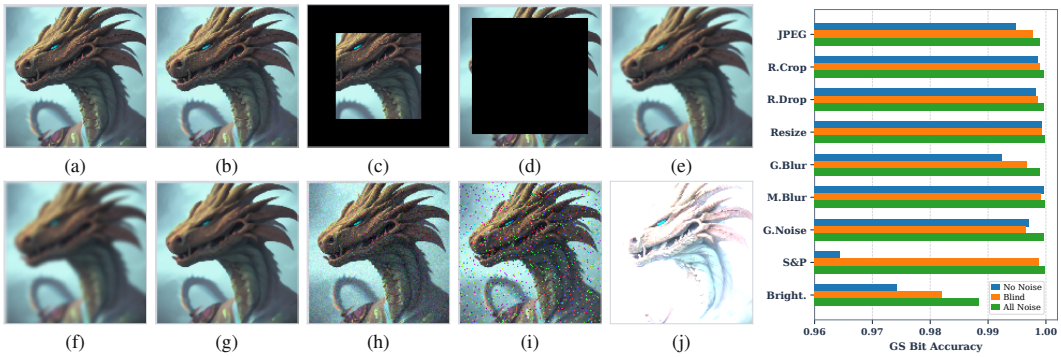


Figure 7: **Robustness and ablation analysis.** Left: Representative examples of the evaluated image distortions. Right: Gaussian Shading bit accuracy under three training settings: No Noise, Blind, and All Noise.

5.3 Generalization

Distortion Severity. To evaluate robustness across distortion strengths, we progressively increase the severity of representative image distortions. As shown in Fig. 6, verification performance gradually decreases as the distortions become stronger, while our method consistently outperforms FARI (Yang et al., 2026) with a larger margin at higher severities. Detailed distortion settings and additional results are provided in Appendix C.4.

Unseen Distortions. We further evaluate generalization to unseen distortions under three training settings: 1) ***No Noise***, without distortion augmentation; 2) ***Blind***, trained with all distortions except the one used for testing; and 3) ***All Noise***, trained with the full distortion set. As shown in Fig. 7 (right), Blind generally outperforms No Noise and approaches All Noise, indicating that robustness learned from other distortions transfers effectively to unseen corruption types.

Cross-Backbone Adaptation. To examine whether the extractor is tied to a specific backbone, we train the VAE-LoRA adapter on SD v1.5 and directly load it into SD v2.1. As shown in Table 2, direct transfer remains effective compared with training from scratch on SD v2.1, and only a few adaptation steps are needed to reach comparable performance. This indicates that the learned mapping captures shared watermark latent structure across model versions rather than inversion behavior specific to one generation trajectory. We further study transfer across SD v3.5 variants of different scales, with results provided in Appendix C.2. These results suggest that, when latent representations are compatible, the extractor can be reused across model versions and scales with little or no retraining, substantially reducing the cost of adapting watermark verification to newly released models.

Table 2: Cross adaptation from SD v1.5 to SD v2.1.

Method	Steps ↓	TR TPR ↑	GS Bit Acc. ↑
Direct Transfer	0	0.997	0.9923
Adaptation	100	0.999	0.9981
From Scratch	1000	0.999	0.9983

5.4 Ablation Study

Watermark Trajectory. To examine the role of different trajectory supervision designs, we progressively ablate their components in Table 3. The simplest baseline directly regresses to the initial watermarked latent z_T^w. Replacing this single target with uniformly sampled intermediate states already improves verification, confirming the benefit of trajectory supervision. Recoverability Select further prioritizes states with stronger watermark evidence, indicating that intermediate states contribute unequally. Finally, magnitude weighting further improves verification by focusing on informative latent components.

Table 3: Ablation on training objectives.

Variant	Adaptive Traj.	Mag.	TR ↑	GS ↑
Initial noise MSE	–	–	0.997	0.9883
Uniform Trajectory	–	–	0.997	0.9924
Recoverability Trajectory	✓	–	0.998	0.9956
+ Magnitude Weighting	✓	✓	0.999	0.9983

LoRA Rank. To study the trade-off between adaptation capacity and parameter efficiency, we vary the LoRA rank of the VAE encoder over $r \in \{1, 4, 8, 16, 32, 64\}$ and report both watermark verification performance and trainable parameters. As shown in Table 4, increasing the rank yields clear gains at low ranks, while performance largely saturates beyond $r = 16$. Rank $r = 8$ already captures most of the improvement with a small parameter overhead, and is therefore used in the remaining experiments.

Table 4: Ablation on LoRA rank.

Rank r	TR TPR ↑	GS Bit Acc. ↑	Params. (M) ↓
1	0.977	0.9962	0.092
4	0.996	0.9974	0.369
8	0.999	0.9983	0.738
16	0.999	0.9985	1.438
32	0.999	0.9985	2.839
64	0.999	0.9988	5.641

5.5 Efficiency and Scalability

Inference and Parameter Efficiency. To assess deployment cost, we compare inference and parameter overhead across extraction methods. As shown in Table 5, Euler and FARI require 160 and 1 denoiser evaluations for verification, whereas our method requires a single VAE encoder forward pass, reducing denoiser NFE to 0 with 19.0 ms per-image verification latency and only 35.01M parameters. This gain comes from removing inversion rather than further compressing it.

Table 5: Efficiency of watermark extraction on SD 3.5.

Method	NFE ↓	Lat. (ms) ↓	Mem. (GiB) ↓	Params. (M) ↓
Euler	160	10042.6	5.09	2503.94
Euler	1	96.2	5.09	2503.94
FARI	1	90.5	5.10	2509.91
ZeroStep	**0**	**19.0**	**0.45**	**35.01**

Large-Scale Verification. To emulate large-scale verification, we increase the query volume from 1 to 10,000 under identical settings. As shown in Fig. 8, runtime grows nearly linearly for both methods, but much more slowly for ours, reaching a $4.91\times$ speedup at 10,000 queries. The gap widens with query volume. By eliminating denoiser evaluations, our method substantially reduces cumulative GPU computation, making it more suitable for high-throughput and large-scale verification.

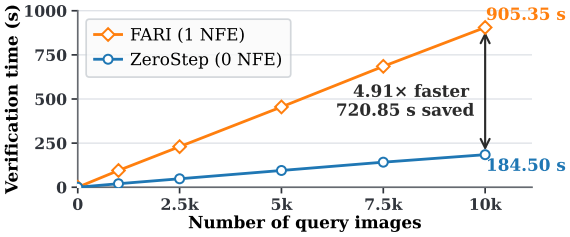


Figure 8: Verification time vs number of query images.

6 Conclusion

In this paper, we revisit the role of diffusion inversion in initial noise watermark verification. Our analysis shows that similarity to the initial watermarked latent decreases substantially along the generation trajectory, while watermark recoverability remains high. This indicates that reliable verification does not require reconstructing the original watermarked noise. Based on this observation, we introduce ZeroStep, which combines Watermark Trajectory Distillation with distortion-robust training to adapt the VAE encoder using LoRA for direct watermark latent extraction. Experiments across SD v1.5, SD v2.1, SDXL-Turbo, and SD v3.5 with Tree-Ring and Gaussian Shading show that our method maintains reliable verification on both clean and distorted inputs, achieves strong gains in accuracy and efficiency, and can be reused across compatible model versions.

REFERENCES

Kasra Arabi, Benjamin Feuer, R. Teal Witter, Chinmay Hegde, and Niv Cohen. Hidden in the noise: Two-stage robust watermarking for images. In *International Conference on Learning Representations*, 2025a.

Kasra Arabi, R Teal Witter, Chinmay Hegde, and Niv Cohen. Seal: Semantic aware image watermarking. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16196–16205. IEEE, 2025b.

Yuwei Chen, Zhenliang He, Jia Tang, Meina Kan, and Shiguang Shan. Osi: One-step inversion excels in extracting diffusion watermarks. *arXiv preprint arXiv:2602.09494*, 2026.

Yuzhuo Chen, Zehua Ma, Han Fang, Weiming Zhang, and Nenghai Yu. Tag-wm: tamper-aware generative image watermarking via diffusion inversion sensitivity. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16723–16732. IEEE, 2025.

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.

Yingying Deng, Xiangyu He, Changwang Mei, Peisong Wang, and Fan Tang. FireFlow: Fast inversion of rectified flow for image semantic editing. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 13110–13128. PMLR, 2025.

Wenbin Duan, Yan Shu, Zhuoyuan Fu, Fangmin Zhao, Yan Li, Yaru Zhao, and Binyang Li. SlerpFlow: Spherical trajectory correction for rectified flow inversion. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 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 (ICCV)*, pp. 22466–22477, October 2023.

Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. Renoise: Real image inversion through iterative noising. In *European Conference on Computer Vision*, pp. 395–413. Springer, 2024.

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

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

Seongmin Hong, Kyeonghyun Lee, Suh Yoon Jeon, Hyewon Bae, and Se Young Chun. On exact inversion of dpm-solvers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7069–7078. IEEE, 2024.

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.

Xiaoxiao Hu, Jiaqi Jin, Sheng Li, Wanli Peng, Xinpeng Zhang, and Zhenxing Qian. Spherical watermark: Encryption-free, lossless watermarking for diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Guanlong Jiao, Biqing Huang, Kuan-Chieh Wang, and Renjie Liao. UniEdit-Flow: Unleashing inversion and editing in the era of flow models. In *International Conference on Learning Representations*, 2026.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Kecen Li, Zhicong Huang, Xinwen Hou, and Cheng Hong. GaussMarker: Robust dual-domain watermark for diffusion models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34688–34701. PMLR, 2025.

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.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

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.

Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick Von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. Lcm-lora: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*, 2023.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6038–6047. IEEE, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, 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 (CVPR)*, pp. 10684–10695, June 2022.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pp. 87–103. Springer, 2024.

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

Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22532–22541. IEEE, 2023.

Fangyikang Wang, Hubery Yin, Yuejiang Dong, Huminhao Zhu, Chao Zhang, Hanbin Zhao, Hui Qian, and Chen Li. Belm: Bidirectional explicit linear multi-step sampler for exact inversion in diffusion models. *Advances in Neural Information Processing Systems*, 37:46118–46159, 2024.

Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Taming rectified flow for inversion and editing. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 64044–64058. PMLR, 2025.

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

Jindong Yang, Han Fang, Weiming Zhang, Nenghai Yu, and Kejiang Chen. FARI: Robust one-step inversion for watermarking in diffusion models. In *International Conference on Learning Representations*, 2026.

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

Zijin Yang, Xin Zhang, Kejiang Chen, Kai Zeng, Qiyi Yao, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading++: rethinking the realistic deployment challenge of performance-lossless image watermark for diffusion models. *arXiv preprint arXiv:2504.15026*, 2025.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024.

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, 2023.

Zhenyu Zhou, Defang Chen, Can Wang, and Chun Chen. Fast ode-based sampling for diffusion models in around 5 steps. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7777–7786. IEEE, 2024.

A Training and Verification Procedures

A.1 Offline Trajectory Targets

For each watermarked training sample, the frozen generator starts from the watermarked initial latent z_T^w, produces the image x^w, and retains the trajectory states $\{z_t^w\}_{t\in\mathcal{T}}$. The original watermark verifier measures the recoverability R_t of each state, while C_t denotes the cosine similarity between z_t^w and z_T^w. Equation 6 combines these quantities into the timestep distribution p_t. During training, we sample $t \sim p_t$ and use z_t^w as the trajectory target. This favors states that have moved away from the initial noise while retaining reliable watermark evidence.

The resulting training set is

$$\mathcal{D}_w = \left\{ \left(x_i^w, z_{T,i}^w, \{z_{t,i}^w\}_{t\in\mathcal{T}} \right) \right\}_{i=1}^{N} \tag{12}$$

The generator is run once to construct $\mathcal{D}_w$, after which the stored targets are reused throughout training. The generator and watermark verifier remain frozen, and neither the trajectory states nor p_t is required at verification time.

A.2 ZeroStep Optimization

Given a sampled target z_t^w, ZeroStep updates only the LoRA parameters ϕ in the VAE encoder. For each training sample, we draw a transformation $\mathcal{A} \sim \mathcal{P}_\mathcal{A}$ and encode the clean and distorted views as

$$\hat{z} = E_\phi(x^w), \qquad \hat{z}^\mathcal{A} = E_\phi(\mathcal{A}(x^w)) \tag{13}$$

For a latent residual $a \in \mathbb{R}^d$, we define the magnitude-weighted squared error as

$$\|a\|_{\bar{w}}^2 = \sum_{j=1}^{d} \bar{w}_j a_j^2, \qquad \bar{w}_j = \frac{|\hat{z}_j|^\gamma}{\sum_{k=1}^{d} |\hat{z}_k|^\gamma} \tag{14}$$

The weights are computed from the clean prediction $\hat{z}$. When $\gamma = 0$, $\bar{w}_j = 1/d$, and the weighted error reduces to the standard mean squared error.

The trajectory loss aligns $\hat{z}$ with the sampled state z_t^w. The distortion loss constrains $\hat{z}^\mathcal{A}$ to remain consistent with $\hat{z}$ while preserving alignment with the initial watermarked latent z_T^w. Algorithm 1 summarizes the resulting optimization procedure.

Algorithm 1 Training ZeroStep

Require: Cached set $\mathcal{D}_w$; timestep distribution p_t; augmentation distribution $\mathcal{P}_\mathcal{A}$
Require: Encoder E_ϕ with trainable LoRA parameters; γ, λ_w, and λ_D
Ensure: Adapted encoder E_ϕ

for each training step **do**
  Sample $(x^w, z_T^w, \{z_s^w\}_{s\in\mathcal{T}}) \sim \mathcal{D}_w$.
  Sample $t \sim p_t$ and $\mathcal{A} \sim \mathcal{P}_\mathcal{A}$.
  $\hat{z} \leftarrow E_\phi(x^w)$; $\quad \hat{z}^\mathcal{A} \leftarrow E_\phi(\mathcal{A}(x^w))$.
  $\bar{w}_j \leftarrow |\hat{z}_j|^\gamma / \sum_{k=1}^{d} |\hat{z}_k|^\gamma, \quad j = 1, \ldots, d.$
  $\mathcal{L}_\mathrm{traj} \leftarrow \|\hat{z} - z_t^w\|_{\bar{w}}^2$.
  $\mathcal{L}_\mathrm{Distort} \leftarrow \|\hat{z}^\mathcal{A} - \hat{z}\|_{\bar{w}}^2 + \lambda_\mathrm{w} \|\hat{z}^\mathcal{A} - z_T^w\|_{\bar{w}}^2$.
  $\mathcal{L} \leftarrow \mathcal{L}_\mathrm{traj} + \lambda_\mathrm{D} \mathcal{L}_\mathrm{Distort}$.
  $\phi \leftarrow \mathrm{AdamStep}(\phi, \nabla_\phi \mathcal{L})$.
end for

A.3 Direct Verification

Given a query image $\hat{x}$ and key K, ZeroStep applies the adapted encoder once and passes its output to the original verifier:

$$\hat{m} = \mathcal{R}_K(\hat{z}) = \mathcal{R}_K(E_\phi(\hat{x})) \tag{15}$$

The denoiser and reverse trajectory are absent from this computation. Consequently, ZeroStep has zero denoiser NFE; its online cost is one VAE-encoder pass and the cost of $\mathcal{R}_K$.

B Implementation and Reproducibility Details

B.1 Training Setup

ZeroStep implementation. We inject rank-8 LoRA modules into the VAE encoder and optimize only the added LoRA parameters. The trajectory targets and timestep sampling distribution in Eq. 6 are precomputed from the frozen generation trajectories and cached throughout training, avoiding repeated generator evaluations during optimization. Both the trajectory and distortion objectives use the magnitude weighting in Eq. 10, where latent components are weighted by the absolute magnitude of the clean encoder prediction. The same weighting is applied consistently to both objectives throughout training.

Baseline reproduction. For solver-based inversion methods, we use the official implementations with their recommended configurations. For learned methods, we use released checkpoints and inference settings whenever available for the target backbone. When reproduction is required, we follow the reported implementation details and use the same training data and distortion protocol whenever applicable. We restrict quantitative comparisons to methods that can be integrated into our unified latent extraction and verification pipeline. OSI Chen et al. (2026) is discussed in the related work but excluded from the quantitative comparison because it reformulates watermark extraction as a task-specific sign classification problem rather than latent recovery. This formulation cannot be directly integrated with watermark schemes such as Tree-Ring and RingID, which require their original latent-space verifiers for detection and multi-key identification, respectively. All evaluated baselines follow the same verification protocol.

Watermark key and message protocol. We follow the original protocol of each watermarking scheme. For Tree-Ring, we follow the original watermark key protocol used in its evaluation, while RingID is additionally evaluated to cover multi-key identification with multiple candidate identities. For Gaussian Shading, a fresh multi-bit watermark message is sampled for each image, together with the corresponding ChaCha20 key and nonce, so the extractor is trained across diverse watermark instances rather than a single fixed pattern. We further evaluate PRC to cover a different multi-bit coding and decoding mechanism based on a fixed secret key. Across all schemes, ZeroStep does not take the watermark key or message as input and modifies only the latent extraction stage, while the original watermark embedder and verifier remain unchanged.

Verification metrics. For Tree-Ring, we report TPR at a stringent fixed FPR of 10^{-3}. Since this operating point lies at the empirical resolution limit of 1,000 negative samples, we estimate it from a fitted ROC curve constructed using 1,000 watermarked and 1,000 unwatermarked examples, rather than relying on a single extreme negative sample. For Gaussian Shading, we report the bit-wise accuracy of the recovered watermark message.

Table 6: Generation and watermark settings for each backbone.

Backbone	Checkpoint	Process	Watermarks	Sampler / steps / CFG	Latent shape
SD v1.5	`runwayml/stable-diffusion-v1-5`	Denoising	Tree-Ring / GS	PNDM / 50 / 7.5	$4 \times 64 \times 64$
SD v2.1	`stabilityai/stable-diffusion-2-1-base`	Denoising	Tree-Ring / GS	PNDM / 50 / 7.5	$4 \times 64 \times 64$
SDXL-Turbo	`stabilityai/sdxl-turbo`	Denoising	Tree-Ring / GS	Euler-A / 1 / 0.0	$4 \times 64 \times 64$
SD v3.5	`stabilityai/stable-diffusion-3.5-medium`	Flow	Tree-Ring / GS	FlowMatchEuler / 40 / 4.5	$16 \times 64 \times 64$

B.2 Watermark and Backbone Configurations

Table 6 specifies the experimental setup used for each backbone in the cross-backbone evaluation. Each backbone is evaluated with both Tree-Ring and Gaussian Shading under its recommended generation setting. SD v1.5 and SD v2.1 follow the standard diffusion configuration, while SDXL-Turbo provides an additional test on a heavily distilled one-step generator. SD v3.5 follows flow-matching dynamics, for which we use Euler inversion as the numerical inversion baseline. Watermark settings are adjusted only to accommodate differences in latent dimensionality: SD v1.5, SD v2.1, and SDXL-Turbo use four-channel latents and retain the original watermark configurations, whereas SD v3.5 uses a 16-channel latent. For Gaussian Shading on SD v3.5, we set $f_{\mathrm{ch}} = 2$ and $f_h = f_w = 8$, yielding a 512-bit payload; for Tree-Ring, watermark embedding is restricted to the last four latent channels. All other embedding and verification settings remain unchanged.

Table 7: Distortion parameters used for training and severity evaluation.

Distortion	Training	Severity sweep (weak → strong)
JPEG (Q)	25	90, 70, 50, 30, 10
Random crop (r)	0.60	0.9, 0.7, 0.5, 0.3, 0.1
Random drop (r)	0.80	0.1, 0.3, 0.5, 0.7, 0.9
Resize (r)	0.25	0.9, 0.7, 0.5, 0.3, 0.1
Gaussian blur (radius)	4	2, 4, 6, 8, 10
Median blur (k)	7	3, 7, 11, 15, 19
Gaussian noise (σ)	0.05	0.02, 0.04, 0.06, 0.08, 0.10
Salt-and-pepper (p)	0.05	0.02, 0.04, 0.06, 0.08, 0.10
Brightness (s)	6	2, 4, 6, 8, 10

Q denotes JPEG quality. For crop and drop, r denotes the side-length ratio of the sampled square region; for resize, r denotes the downsampling ratio. k denotes the median-filter kernel size, and p denotes the corruption probability for salt-and-pepper noise. Gaussian noise is zero-mean. Crop retains the sampled square region and zeros the remainder, whereas drop zeros the sampled region. Resized images are restored to the original resolution using bilinear interpolation.

B.3 Distortion Settings

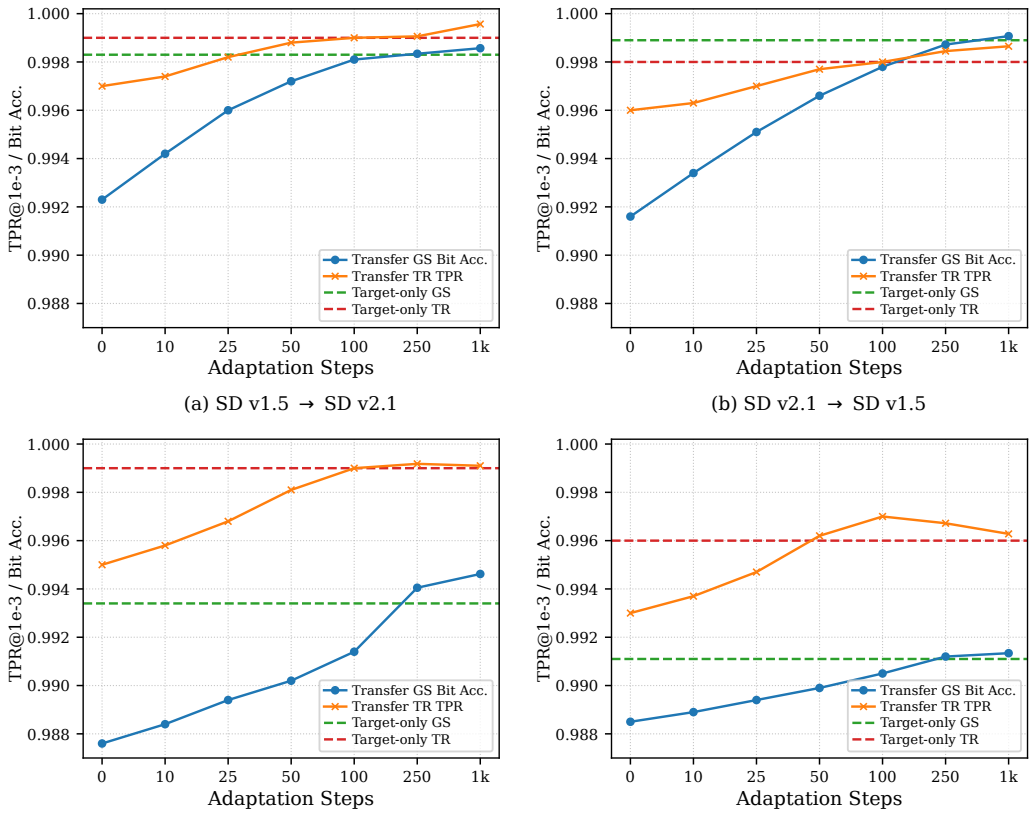
Table 7 summarizes the distortion settings used for robustness training and evaluation. The nine transformations cover common image degradations, including compression, spatial corruption, filtering, additive noise, and brightness variation. During training, one distortion is sampled uniformly for each image and applied at a fixed moderate strength, while the corresponding clean view is retained to enforce consistency between the clean and distorted representations. Distortions are applied individually rather than composed. For the main robustness evaluation, each distortion is applied independently at the moderate strength listed in the "Training" column of Table 7. We additionally conduct a severity sweep in which the strength of each distortion is varied over five levels while the remaining evaluation protocol is kept fixed. Figure 6(a) reports verification performance averaged across all distortions at each severity level, while Figures 6(b–j) report the corresponding results for individual distortion types. All compared methods are evaluated on the same corrupted queries under each setting.

Table 8: Comparison with direct verification using the original frozen VAE encoder. Both Frozen VAE and ZeroStep require zero denoiser NFE. "Frozen VAE" directly feeds the pretrained VAE endpoint latent to the original watermark verifier without any adaptation.

		Tree-Ring		Gaussian Shading	
DM	**Methods**	**Clean TPR ↑**	**Avg. TPR ↑**	**Clean Bit Acc. ↑**	**Avg. Bit Acc. ↑**
SD v1.5	Frozen VAE	0.820	0.219	0.9406	0.8757
	ZeroStep	1.000	0.998	1.0000	0.9989
SD v2.1	Frozen VAE	0.866	0.235	0.9373	0.8692
	ZeroStep	1.000	0.999	1.0000	0.9983

C Additional Generalization and Robustness

We further examine the generalization of our method beyond the default training and evaluation setting. We first compare against a training-free frozen-VAE baseline to isolate the effect of encoder adaptation and better understand how much watermark evidence is already preserved in the pretrained latent representation. We then consider adapter transfer across compatible model variants, changes in prompt distributions, and distortions excluded during training. Together, these analyses evaluate both the role of encoder adaptation and the robustness of the learned extractor under model, data, and corruption shifts.



(a) SD v1.5 → SD v2.1

(b) SD v2.1 → SD v1.5

(c) SD v3.5M → SD v3.5L

(d) SD v3.5L → SD v3.5M

Figure 9: Adapter transfer across compatible model variants. (a) SD v1.5 → SD v2.1. (b) SD v2.1 → SD v1.5. (c) SD v3.5 Medium → SD v3.5 Large. (d) SD v3.5 Large → SD v3.5 Medium. Solid curves show transferred-adapter performance over target-model adaptation steps, while dashed lines denote the corresponding target-only references trained for 1,000 steps.

C.1 Frozen VAE Zero-Step Baseline

To separate the benefit of removing inversion from that of encoder adaptation, we evaluate a training-free zero-step baseline that directly feeds the latent produced by the original frozen VAE encoder into the watermark verifier. This baseline requires neither diffusion inversion nor additional training.

As shown in Table 8, the frozen VAE already preserves non-trivial watermark information, consistent with our observation that watermark evidence remains at the generation endpoint. However, ZeroStep consistently improves verification performance, particularly under image distortions. This comparison shows that removing inversion alone is insufficient, and that the learned VAE adaptation is important for extracting a robust, watermark-compatible representation.

C.2 Transfer across Compatible Model Variants

We examine whether a ZeroStep adapter can be reused across model variants with compatible VAE representations. We consider bidirectional transfer between SD v1.5 and SD v2.1, as well as between SD v3.5 Medium and Large. In each direction, the source adapter is first applied directly to the target model and then further adapted using target-model trajectories, while target-only training for 1,000 steps serves as the reference. Figures 9(a–d) show the full adaptation curves, and Tables 9 and 10 summarize representative operating points.

For SD v1.5 and SD v2.1, direct transfer already preserves strong verification performance in both directions. SD v1.5 → SD v2.1 achieves a Tree-Ring TPR of 0.997 and a Gaussian Shading bit accuracy of 0.9923 without any target-model training, while the reverse transfer reaches 0.996 and 0.9916, respectively. After only 100 target-model steps, the Gaussian Shading accuracy increases to 0.9981 and 0.9978, approaching the corresponding target-only results of 0.9983 and 0.9989. Tree-

Ring reaches 0.999 and 0.998 after adaptation, matching the target-only references. The full curves in Figures 9(a–b) further show that most of the improvement is obtained during the early adaptation stage, with later updates producing only modest gains. This indicates that the transferred adapter already provides a strong initialization and requires only limited target-specific adjustment.

Table 9: Cross-version transfer between SD v1.5 and SD v2.1. Target-only training uses no transferred adapter and trains directly on the target model for 1,000 steps.

Transfer	Setting	Target steps	TR TPR ↑	GS Bit Acc. ↑
SD v1.5 → SD v2.1	Direct transfer	0	0.997	0.9923
	Adapted transfer	100	0.999	0.9981
	Target-only training	1000	0.999	0.9983
SD v2.1 → SD v1.5	Direct transfer	0	0.996	0.9916
	Adapted transfer	100	0.998	0.9978
	Target-only training	1000	0.998	0.9989

We next consider SD v3.5 Medium and Large, which share the same VAE but differ in the scale of the generative model. This setting isolates the effect of generator-side changes while keeping the VAE architecture and latent space fixed. Direct Medium → Large transfer achieves 0.995 Tree-Ring TPR and 0.9876 Gaussian Shading bit accuracy, which increase to 0.999 and 0.9914 after 100 target-model updates, compared with target-only values of 0.999 and 0.9934. In the reverse direction, Large → Medium starts from 0.993 TPR and 0.9885 bit accuracy and reaches 0.997 and 0.9905 after 100 steps, close to the target-only results of 0.996 and 0.9911. Figures 9(c–d) show a similar saturation behavior to that observed for SD v1.5/v2.1: adaptation improves rapidly during the first few hundred updates and then largely stabilizes around the target-only reference.

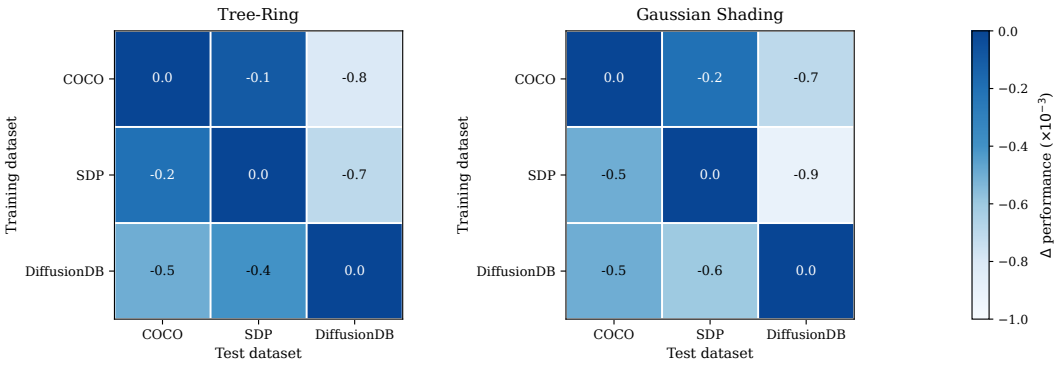
Table 10: Cross-scale transfer between SD v3.5 Medium and Large. Target-only training denotes training directly on the target model without using a transferred adapter.

Transfer	Setting	Target steps	TR TPR ↑	GS Bit Acc. ↑
Medium → Large	Direct transfer	0	0.995	0.9876
	Adapted transfer	100	0.999	0.9914
	Target-only training	1000	0.999	0.9934
Large → Medium	Direct transfer	0	0.993	0.9885
	Adapted transfer	100	0.997	0.9905
	Target-only training	1000	0.996	0.9911

These results show that the learned adapter can be reused across model variants with compatible VAE representations. Direct transfer retains most of the verification capability, while a short adaptation stage is sufficient to compensate for changes in the target model's generation trajectory and recover performance close to training directly on the target model. The results therefore support adapter reuse across both model versions and model scales, reducing the need to retrain the extractor from scratch for every compatible variant. We do not transfer adapters between the SD v1.x/2.x family and SD v3.5 because their VAE architectures and latent dimensionalities differ, making direct reuse of the same LoRA parameters infeasible.

C.3 Generalization across Datasets

We further examine the dependence of the learned extractor on the training prompt distribution. We fix the generative model to SD v2.1 and consider three prompt datasets: MS-COCO, Stable-Diffusion-Prompts (SDP), and DiffusionDB. An adapter is trained on 1,000 prompts from each dataset and evaluated, without further adaptation, on 1,000 disjoint prompts from all three datasets, while keeping the generation and watermark settings fixed.



(a) Tree-Ring (b) Gaussian Shading

Figure 10: Cross-dataset generalization on SD v2.1. Each cell reports the performance change relative to the corresponding matched train–test setting. Left: Tree-Ring TPR. Right: Gaussian Shading bit accuracy.

Figure 10 reports the performance change of each train–test pair relative to its corresponding in-domain result, with negative values indicating degradation. The diagonal entries are therefore zero by construction, while the off-diagonal entries quantify the cross-dataset transfer gap. Overall, the gaps remain very small for both watermarking schemes. Tree-Ring TPR changes by at most 0.8×10^{-3}, while Gaussian Shading bit accuracy changes by at most 0.9×10^{-3}. The largest numerical drops generally occur when evaluating on DiffusionDB, but all cross-dataset differences remain below 10^{-3}. These results indicate that the learned extractor is largely insensitive to shifts in the training prompt distribution.

C.4 Generalization to Unseen Distortions

Table 11: Leave-one-distortion-out generalization. Each entry reports Blind / All Aug.; Blind excludes the test distortion during training.

Held-out distortion	**SD v2.1**		**SD v3.5**	
	TR TPR ↑	**GS Bit Acc. ↑**	**TR TPR ↑**	**GS Bit Acc. ↑**
JPEG	0.999 / 1.000	0.9978 / 0.9990	0.997 / 1.000	0.9898 / 0.9924
Random crop	0.999 / 1.000	0.9990 / 0.9997	0.999 / 1.000	0.9986 / 0.9995
Random drop	0.998 / 1.000	0.9987 / 0.9996	0.996 / 1.000	0.9938 / 0.9963
Resize	0.999 / 1.000	0.9993 / 0.9999	0.998 / 1.000	0.9899 / 0.9926
Gaussian blur	0.997 / 0.999	0.9968 / 0.9990	0.974 / 0.989	0.9527 / 0.9619
Median blur	0.999 / 1.000	0.9992 / 0.9999	0.993 / 0.999	0.9912 / 0.9941
Gaussian noise	0.981 / 0.989	0.9965 / 0.9996	0.960 / 0.980	0.9939 / 0.9970
Salt-and-pepper	0.996 / 0.999	0.9988 / 0.9999	0.988 / 0.996	0.9857 / 0.9905
Brightness	0.999 / 1.000	0.9821 / 0.9884	0.997 / 1.000	0.9928 / 0.9954
Average	0.996 / 0.999	0.9965 / 0.9983	0.989 / 0.996	0.9876 / 0.9911

We extend the unseen-distortion analysis in Sec. 5.3 with a complete leave-one-distortion-out evaluation. For each test distortion, the *Blind* model is trained on the remaining eight distortions, whereas *All Aug.* uses the full distortion set. This setting isolates whether robustness learned from other corruptions transfers to a distortion never observed during training. Table 11 reports the results for all nine distortions on both SD v2.1 and SD v3.5.

Blind remains close to All Aug. for most held-out distortions on both backbones. The largest gaps for Tree-Ring appear under Gaussian blur and Gaussian noise on SD v3.5, where removing the corresponding augmentation reduces TPR by 0.015 and 0.020, respectively. Other distortions transfer more readily, with only minor differences between Blind and All Aug. Overall, the results across Tree-Ring and Gaussian Shading indicate that robustness learned from the remaining distortions transfers effectively to unseen corruption types.

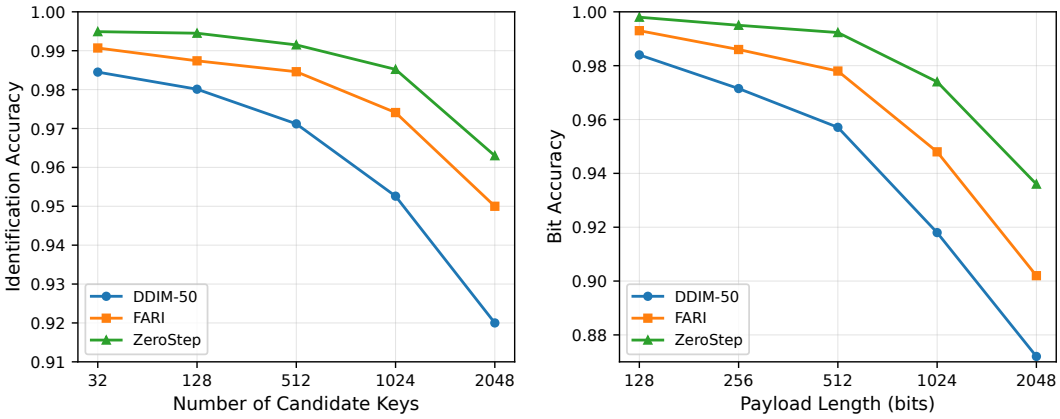


Figure 11: Scaling behavior on additional watermarking schemes. Left: distortion-averaged RingID top-1 identification accuracy as the number of candidate keys increases from 32 to 2,048. Right: distortion-averaged PRC bit accuracy as the payload length increases from 128 to 2,048 bits.

Table 12: Robustness on additional watermarking schemes using SD v2.1. RingID reports top-1 identification accuracy over 32 candidate keys, while PRC reports bit accuracy for a 512-bit payload.

WM	Method	NFE	Clean	Avg.	Jpeg	R.Crop	R.Drop	Resize	G.Blur	M.Blur	G.Noise	S&P	Bright
Top-1 Identification Accuracy of RingID (32 candidate keys)													
RingID	DDIM	50	1.0000	0.9850	0.9980	0.9750	0.9710	0.9910	0.9900	0.9970	0.9780	0.9820	0.9830
	DDIM	1	0.9940	0.9658	0.9890	0.9510	0.9460	0.9780	0.9750	0.9890	0.9510	0.9580	0.9550
	FARI	1	0.9990	0.9907	0.9980	0.9870	0.9840	0.9950	0.9950	0.9990	0.9850	0.9900	0.9830
	ZeroStep	0	**1.0000**	**0.9949**	**0.9990**	**0.9910**	**0.9890**	**0.9970**	**0.9980**	**0.9990**	**0.9920**	**0.9950**	**0.9940**
Bit Accuracy of PRC Watermark (512-bit payload)													
PRC	DDIM	50	0.9982	0.9571	0.9825	0.9460	0.9390	0.9900	0.9700	0.9940	0.9300	0.9150	0.9470
	DDIM	1	0.9925	0.9343	0.9680	0.9150	0.9030	0.9780	0.9450	0.9850	0.9050	0.8890	0.9210
	FARI	1	0.9990	0.9780	0.9920	0.9730	0.9680	0.9960	0.9880	0.9970	0.9630	0.9550	0.9700
	ZeroStep	0	**0.9998**	**0.9923**	**0.9980**	**0.9910**	**0.9880**	**0.9985**	**0.9960**	**0.9993**	**0.9860**	**0.9820**	**0.9920**

D Applicability to Additional Watermarking Schemes

To examine whether ZeroStep is tied to a particular watermark construction, we additionally evaluate RingID Ci et al. (2024) and PRC Gunn et al. (2025) on SD v2.1. RingID introduces multi-key identification, while PRC uses a different multi-bit coding and decoding mechanism. For each scheme, we retain its original watermark embedding and verification procedure and train ZeroStep using the same trajectory-based framework and distortion protocol as in the main experiments. The trajectory recoverability R_t in Eq. 6 is defined using the verification metric of the corresponding watermark scheme.

D.1 RingID: Multi-Key Identification

For RingID, we use 32 candidate keys and report top-1 identification accuracy under the same clean and distorted queries as in the main experiments. As shown in Table 12, all methods remain strong under clean inputs and common image distortions, while ZeroStep achieves the highest distortion-averaged identification accuracy without diffusion inversion. The advantage is more apparent under spatial perturbations and noise corruptions, where inversion-based extraction introduces larger degradation. These results indicate that the directly extracted latent preserves the key-specific information required by the original RingID verifier after image distortion.

We further vary the number of candidate keys from 32 to 2,048 while keeping the extractor fixed. As shown in Figure 11(left), identification accuracy changes only slightly when the candidate set grows from 32 to 128 keys and remains relatively stable up to 512 keys. The degradation becomes more

pronounced beyond 512 keys as the identification task becomes increasingly difficult. ZeroStep exhibits a slower decline than the inversion-based alternatives and retains the highest accuracy at 2,048 candidate keys.

D.2 PRC: Multi-Bit Recovery

For PRC, we use a 512-bit payload and report bit accuracy, defined as the fraction of correctly recovered payload bits. Table 12 shows that ZeroStep retains high recovery accuracy across the same distortion set, achieving a distortion-averaged bit accuracy of 0.9923. The advantage over inversion-based extraction is more visible under crop, drop, and additive noise, where image perturbations lead to larger recovery errors for the inversion baselines. This extends the multi-bit evaluation beyond Gaussian Shading to a watermark with a different coding structure.

We further vary the payload length from 128 to 2,048 bits under the same extraction configuration. Figure 11(right) shows that recovery accuracy already begins to decrease as the payload grows from 128 to 256 bits, followed by a larger degradation beyond 512 bits. This trend becomes substantially stronger at 1,024 and 2,048 bits, where the increased payload makes accurate recovery more challenging. ZeroStep degrades more gracefully than the inversion-based alternatives and maintains the highest bit accuracy throughout the evaluated payload range.

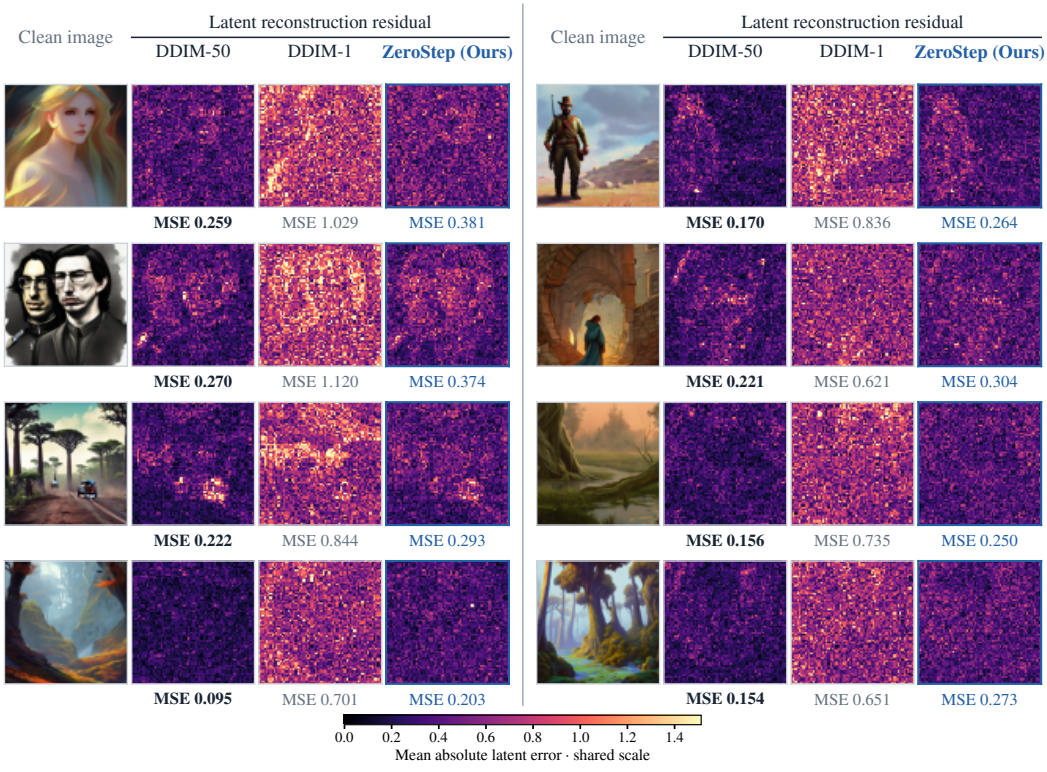


Figure 12: Representative latent reconstruction residuals on clean SD v2.1 queries. For each query, we show the clean image and channel-averaged absolute residuals $\frac{1}{C}\sum_c |\hat{z}_c - z^w_{T,c}|$ for DDIM-50, DDIM-1, and ZeroStep. All residual maps share the same color scale, with the corresponding latent MSE reported below each map.

E Reconstruction Fidelity versus Watermark Verification

Our analysis in Sec. 3 shows that watermark evidence can remain recoverable even after the latent has substantially deviated from the initial watermarked noise. Here, we examine the same question from the extraction perspective: whether a representation must accurately reconstruct z^w_T to support reliable watermark verification.

E.1 Reconstruction and Verification Mismatch

We compare DDIM-50, DDIM-1, and ZeroStep on clean SD v2.1 queries. For each method, we measure the mean per-element MSE between the extracted latent $\hat{z}$ and the original watermarked latent z_T^w, together with Tree-Ring and Gaussian Shading verification performance. The results are summarized in Table 13.

Table 13: Initial noise reconstruction and watermark verification on clean SD v2.1 queries. Reconstruction error is measured against the original watermarked latent z_T^w.

Method	**NFE**	**MSE** $\downarrow$	**TR TPR** $\uparrow$	**GS Bit Acc.** $\uparrow$
DDIM	50	0.250	1.000	1.0000
DDIM	1	0.896	1.000	0.9987
ZeroStep	0	0.347	1.000	1.0000

The reconstruction errors differ substantially despite nearly saturated watermark verification. DDIM-50 most closely reconstructs the original watermarked noise, with an MSE of 0.250. ZeroStep exhibits a noticeably larger reconstruction error of 0.347, yet retains saturated verification for both watermarking schemes. In contrast, reducing DDIM inversion to a single step causes the reconstruction error to increase sharply to 0.896, while its verification performance remains nearly unchanged on clean queries. These results reveal a clear mismatch between global latent reconstruction fidelity and watermark verification: accurately reproducing the full initial watermarked noise can improve latent similarity, but is not required for reliable verification.

E.2 Latent Error Visualization

We further visualize the reconstruction–verification mismatch at the latent level. Given an extracted latent $\hat{z}$, we compute the channel-averaged absolute residual $\frac{1}{C}\sum_c |\hat{z}_c - z^w_{T,c}|$. Figure 12 shows representative clean queries together with residual maps from DDIM-50, DDIM-1, and ZeroStep. All residual maps use the same color scale, and the per-sample MSE with respect to z_T^w is reported below each map.

The visualization is consistent with the aggregate results in Table 13. DDIM-50 generally produces the smallest residuals and therefore most closely reconstructs the original watermarked latent. DDIM-1 exhibits substantially larger and more spatially widespread errors, reflecting the loss of reconstruction fidelity when inversion is reduced to a single step. ZeroStep lies between these two regimes: its residuals are consistently larger than those of DDIM-50, but substantially smaller than those of DDIM-1 in the displayed examples. Importantly, this additional deviation from z_T^w relative to DDIM-50 does not reduce watermark verification performance. These observations further indicate that successful verification does not require recovering the full initial latent with high fidelity; preserving the watermark-relevant information is sufficient.