BEYOND INVERSION: SOURCE-ANCHORED FORGERY ATTACKS ON SEMANTIC WATERMARKS FOR DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Semantic watermarking encodes provenance signals in the initial noise of diffusion models without modifying model weights, enabling source attribution that is robust to common image transformations. Yet, malicious semantic edits can preserve these signals, allowing manipulated content to retain attribution to the original generator or user. Inversion-based forgery attacks first estimate an initial noise latent in an attacker-controlled model and then regenerate the image under a target prompt. Watermark retention along this path, however, is sensitive to mismatch between the attacker and victim models. We propose an inversion-free, source-anchored forgery attack that bypasses this reconstruction step. Using the observed image's latent encoding as a fixed anchor, the attack integrates differences between velocity predictions at paired source and target states to perform prompt-guided semantic manipulation. It requires neither the source prompt nor queries to the victim generator or watermark verifier. Experiments across five victim generators and multiple semantic watermark methods demonstrate consistently higher watermark retention than inversion-based baselines. The attack also produces semantic forgeries that pass watermark verification under a defense designed to bind watermarks to image semantics. These results expose a gap between robust source attribution and protection against semantic forgery.

1 INTRODUCTION

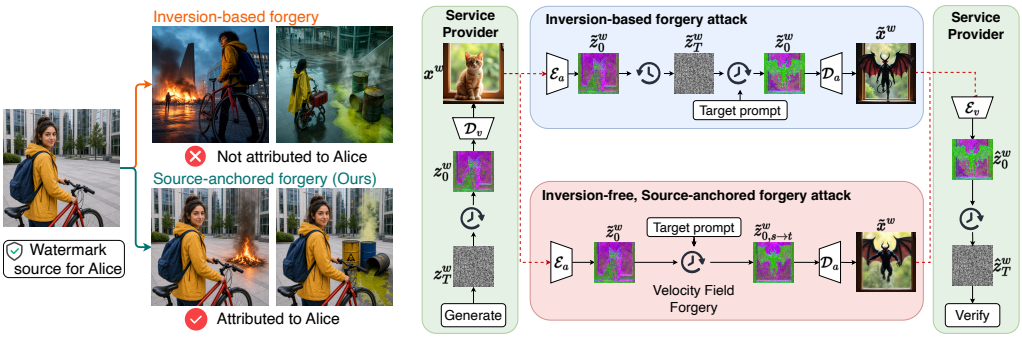
Recent advances in generative AI, exemplified by diffusion models (Ho et al., 2020; Song et al., 2021; Rombach et al., 2022; Peebles & Xie, 2023), have rapidly improved image synthesis, producing images with striking realism and creative diversity. These capabilities have made visual creation more accessible, but they have also lowered the barrier to fabricating deceptive or harmful content (Vaccari & Chadwick, 2020; Mirsky & Lee, 2021). As synthetic images become increasingly difficult to distinguish from authentic ones, reliable provenance mechanisms are urgently needed for content governance and accountability (Bird et al., 2023; Jaidka et al., 2025; Ren et al., 2024).

Watermarking has emerged as a primary means of tracing AI-generated content (Fernandez et al., 2023; Wen et al., 2023; Yang et al., 2024). Among diffusion-specific approaches, semantic watermarks encode structured watermark information into the initial latent used for image generation. Such watermarks require no modification to generators' weights and are robust to common image transformations (Wen et al., 2023; Yang et al., 2024).

However, robustness to benign perturbations does not ensure security against semantic manipulation, in which an attacker deliberately changes the depicted content. As Figure 1(a) illustrates, an attacker may alter the meaning of a watermarked image while retaining the source watermark, causing malicious or forged content to be falsely attributed to the original generator or user. Existing black-box forgery attacks against semantic watermarks typically pursue this false-attribution objective through inversion: they recover a model-specific initial latent from the observed image and regenerate it under a target prompt (Müller et al., 2025). We refer to these methods as *inversion-based forgery attacks*.

Inversion-based forgery requires source-watermark evidence to survive reconstruction and regeneration within a proxy model. This process is sensitive to model mismatch: when the proxy and

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.



(a) Potential false attribution (b) Attack-path comparison

Figure 1: Semantic-watermark forgery and attack paths. (a) Conceptual illustration of semantic manipulation: a tampered image can retain the source watermark, creating a false-attribution risk when verification is interpreted as authentication of the tampered content. (b) Provider-side generation and verification surround two attack paths: inversion-based forgery reconstructs an attacker-space initial latent before target-prompt regeneration (top), whereas our source-anchored attack directly transports the observed-image latent toward target semantics (bottom).

victim use different generative model families, the recovered latent and regeneration path may no longer preserve the victim-side watermark structure. We refer to this cross-model failure mode as the *inversion bottleneck*. The problem is especially relevant in black-box settings, where the victim model is unavailable. This raises a central question: can an attacker alter image semantics while retaining the source watermark, without reconstructing how the victim generated the source image?

In this work, we formulate an inversion-free, source-anchored forgery attack against semantic watermarks, mitigating watermark-evidence loss under model mismatch. Starting from the observed image encoding, the attack integrates target-conditioned displacements without recovering the victim's initial latent or generation trajectory. By anchoring the transport at the observed image, the attack bypasses trajectory recovery. Experiments across five victim generators show stronger source-watermark retention under the reported configurations. Independent alignment diagnostics show that the stronger source preservation accompanies smaller target-prompt shifts. We use paired evidence-loss comparisons to quantify model-family sensitivity and interpret detector acceptance separately from source fidelity. The resulting study examines how editing paths affect provenance signals under the evaluated configurations.

In summary, our main contributions are as follows: (1) We characterize the cross-model *inversion bottleneck* through a paired victim–attacker family crossover, measuring the inversion path's sensitivity to model pairing. (2) We formulate an inversion-free, source-anchored forgery under an unknown-victim setting with a fixed attacker, empty source conditioning, and no detector queries. (3) We demonstrate improved watermark retention across multiple victim generators and semantic watermark methods, and show that targeted semantic forgeries remain accepted under a defense designed to bind watermarks to image semantics.

2 Related Work

Diffusion Models and Inversion. Diffusion models synthesize images through iterative denoising (Ho et al., 2020). Latent Diffusion Models (LDMs) (Rombach et al., 2022) perform this process in the compact latent space of a pretrained autoencoder: an encoder $\mathcal{E}$ maps an image x to its latent representation $z_0 = \mathcal{E}(x) \in \mathbb{R}^{h\times w\times c}$, and a decoder $\mathcal{D}$ reconstructs $x' = \mathcal{D}(z_0)$. Given a text condition p, generation maps initial noise $z_T \sim \mathcal{N}(0, I)$ to a clean latent $z_0 = \mathcal{G}_{T\to 0}(z_T; p)$. DDIM (Song et al., 2021) defines a deterministic sampling trajectory, while DDIM inversion (Mokady et al., 2023) reverses the sampling updates to estimate $\hat{z}_T = \mathcal{I}_{0\to T}(z_0; p)$. The recovered state is model-specific because the generation and inversion traversals depend on the autoencoder, prediction parameterization, and transport dynamics.

Watermarking for Diffusion Models. Post-processing watermarking embeds signals into generated images using classical (Cox et al., 2007; Al-Haj, 2007; Navas et al., 2008; Li et al., 2009) or learned encoder–decoder methods (Zhu et al., 2018; Tancik et al., 2020; Zhang et al., 2019; Bui et al., 2023). Another family integrates watermarking into generation by fine-tuning model components or weights (Fernandez et al., 2023; Feng et al., 2024; Rezaei et al., 2025; Fei et al., 2025). Semantic watermarks instead encode watermark structure in the initial latent while keeping the pretrained generator fixed. In the two outer panels (green) of Figure 1(b), a watermark-bearing latent z_T^w produces $x^w = \mathcal{D}_v(\mathcal{G}_{v,T\to 0}(z_T^w; p))$, while the verifier recovers $\hat{z}_{v,T}^w = \mathcal{I}_{v,0\to T}(\mathcal{E}_v(x^w))$ from the image under verification and decodes the watermark from this latent. Tree-Rings (Wen et al., 2023) uses a Fourier-domain ring pattern, whereas Gaussian Shading (Yang et al., 2024) encodes encrypted messages through probability-preserving Gaussian bins. PRC Watermark (Gunn et al., 2025) selects initial latents through pseudorandom error-correcting codes, and Spherical Watermark (Hu et al., 2026) maps high-entropy codes to Gaussian-distributed initial noise without per-image encryption. These methods differ in latent-code construction and verification, but share training-free integration and robust detection after common image perturbations.

Watermark Forgery Attacks and Defenses. Watermark forgery attacks seek false attribution by producing new or manipulated content accepted by a source-watermark verifier. Cover-based attacks transplant watermark information onto an independent image. WMCopier (Dong et al., 2025) learns a watermark distribution from multiple watermarked images and injects it into covers through shallow inversion and iterative refinement. PnP (Zhu et al., 2025) inverts a watermarked reference to estimate $\tilde{z}_T^w$, then uses it to initialize a regenerative diffusion model conditioned on the visual and textual features of a cover image. Imprint-Forgery (Müller et al., 2025) instead optimizes each cover latent so that its proxy-model inversion approaches the estimated watermarked latent while preserving cover content. In the cover-free setting, Reprompting (Müller et al., 2025) inverts a watermarked reference in an unrelated proxy model to recover its watermarked initial noise, and then regenerates an image under a target prompt. In this work, we focus on the cover-free, target-directed forgery setting, while removing its dependence on initial-noise recovery from the forgery path. Figure 1(b) details the attack path of inversion-based reprompting in the upper central panel (blue). It encodes the observed watermarked image as $\tilde{z}_0^w = \mathcal{E}_a(x^w)$, estimates $\tilde{z}_T^w = \mathcal{I}_{a,0\to T}(\tilde{z}_0^w; p^\emptyset)$, and regenerates $\tilde{x}^w = \mathcal{D}_a(\mathcal{G}_{a,T\to 0}(\tilde{z}_T^w; p^{\mathrm{tar}}))$. Forgery success is therefore coupled to the attacker's full inversion–generation pipeline: the embedded watermark structure must be recovered by inversion and remain intact during target-conditioned regeneration. This dependency can become fragile when victim and attacker differ substantially in architecture or generation dynamics. By contrast, the lower central panel (red) shows our method anchoring the process at $z_0^{\mathrm{src}} = \mathcal{E}_a(x^w)$ and integrating target-conditioned velocity displacements directly from this observed-image latent. It bypasses initial-noise estimation and the subsequent regeneration used in inversion-based reprompting.

On the defense side, SemBind (Zhang et al., 2026) defends semantic watermarks against forgery attacks by binding latent watermarks to image-derived semantic codes. We investigate whether this defense withstands our source-anchored forgery attack.

3 Methodology

3.1 Threat Model and Attack Objective

We present a black-box forgery attack against semantic watermarks in diffusion-generated images. Given a watermarked image x^w, the attacker uses a target prompt p^{tar} to guide an attacker-controlled proxy model in tampering with the image's semantic content. The victim generator, source prompt p_s, watermark key k, initial noise, and generation trajectory are unavailable. The attacker makes no queries to the victim generator or its watermark verifier. We express the attack as

$$y = \mathcal{A}_a(x^w, p^{\mathrm{tar}}), \tag{1}$$

where a denotes the attacker-controlled proxy model and attack procedure, which remain fixed across victim models without conditioning on their identities.

A successful forgery produces a tampered image y that realizes the target semantic manipulation while retaining the watermark associated with x^w under victim-side verification. The attack modifies the observed source image, allowing parts of its content to remain while other content is altered or

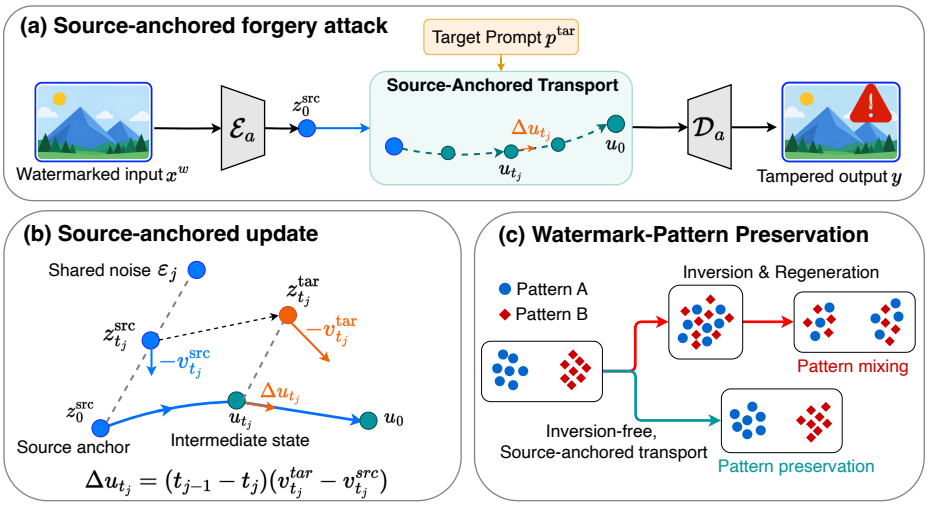


Figure 2: Framework overview. (a) Source-anchored forgery pipeline. The observed watermarked image is encoded once, transported under the target prompt, and decoded into a tampered image. (b) Shared-noise paired states and source-anchored transport. The orange arrows at u_{t_j} denote the state displacement Δu_{t_j}. (c) Schematic comparison of watermark-pattern mixing under inversion–regeneration and preservation under source-anchored transport. Blue circles and red diamonds represent distinct watermark patterns.

new semantic information is introduced according to p^{tar}. The retained watermark creates a false-attribution risk when verification is interpreted as authenticating the manipulated content.

3.2 Framework Overview

Source-anchored forgery. Figure 2(a) illustrates the attack framework for the forgery mapping in Eq. (1). The observed watermarked image is encoded once into the proxy model's latent space. The central stage, *source-anchored latent transport*, accumulates target-directed displacements relative to this source encoding, bypassing the initial-noise reconstruction required by inversion-based reprompting. Inspired by flow-based editing (Kulikov et al., 2025), we construct source-anchored displacements from the difference between the velocities predicted by the proxy model at paired source and target states. The source branch is built from the source encoding and uses an empty source prompt $p^{\emptyset}$. The target branch incorporates the accumulated displacement and uses p^{tar}. Thus, source content enters through the image encoding, while the target prompt specifies the intended semantic change. The two states share the same noise, and the resulting velocity difference is integrated to update the intermediate latent. Finally, after the paired updates, the tampered image is obtained by directly decoding the terminal latent z_0^{tar}.

Paired conditional updates. Let $\mathcal{E}_a$ and $\mathcal{D}_a$ denote the flow-based proxy's encoder and decoder, and $V_a(z, t, p)$ its velocity-field function at latent z and time t, conditioned on prompt p. We encode $z_0^{\text{src}} = \mathcal{E}_a(x^w)$ once and retain it as a fixed source reference.

We use a discrete time sequence $0 = t_0 < t_1 < \cdots < t_J \leq 1$ from the proxy's sampling schedule, traversed from t_J to 0. Here t denotes the normalized flow-time variable that determines the Gaussian-noise weight in the source-state interpolation. Initialize $u_{t_J} = z_0^{\text{src}}$ and denote the endpoint by $z_0^{\text{tar}} = u_0$. The zero subscripts in z_0^{src} and z_0^{tar} refer to the clean endpoint $t = 0$. The intermediate state u_t evolves directly between these endpoints, while auxiliary noisy states supply inputs for velocity prediction.

Figure 2(b) illustrates the paired-state construction underlying source-anchored transport. At each step $j = J, \ldots, 1$, independently draw $\epsilon_j \sim \mathcal{N}(0, I)$ and construct the paired noisy states

$$z_{t_j}^{\text{src}} = (1 - t_j) z_0^{\text{src}} + t_j \epsilon_j, \qquad z_{t_j}^{\text{tar}} = z_{t_j}^{\text{src}} + (u_{t_j} - z_0^{\text{src}}). \tag{2}$$

Each step reuses the source reference z_0^{src}, while the target state adds the accumulated displacement $u_{t_j} - z_0^{\mathrm{src}}$. The source and target velocity vectors are

$$v_{t_j}^{\mathrm{src}} = V_a(z_{t_j}^{\mathrm{src}}, t_j, p^{\emptyset}), \qquad v_{t_j}^{\mathrm{tar}} = V_a(z_{t_j}^{\mathrm{tar}}, t_j, p^{\mathrm{tar}}). \tag{3}$$

Both vectors are outputs of the same proxy model under different states and prompts. Their difference $\Delta v_{t_j} = v_{t_j}^{\mathrm{tar}} - v_{t_j}^{\mathrm{src}}$ defines the transport velocity for this step. We apply the Euler update

$$\Delta u_{t_j} = (t_{j-1} - t_j)\Delta v_{t_j}, \qquad u_{t_{j-1}} = u_{t_j} + \Delta u_{t_j}. \tag{4}$$

The displacement Δu_{t_j} shown in Figure 2(b) advances the state from t_j to t_{j-1} and includes the negative time increment. After J updates, decoding $y = \mathcal{D}_a(z_0^{\mathrm{tar}})$ completes the pipeline in Figure 2(a). Algorithm 1 summarizes the computation. Appendix A clarifies the time and conditioning conventions and derives the accumulated source-relative displacement.

3.3 Mechanism Analysis

Watermark-pattern correspondence. Semantic watermarks encode information in the initial noise latent. Gaussian Shading, for example, samples latent coordinates from Gaussian intervals determined by randomized message symbols while preserving the marginal noise distribution (Yang et al., 2024). The verifier estimates this latent from the image and decodes the watermark. Watermark retention therefore requires sufficient agreement with the source message, while image quality and a matching noise distribution alone do not guarantee this agreement. Figure 2(c) schematically contrasting pattern mixing under inversion–regeneration with the pattern-preservation rationale behind our source-anchored design.

Algorithm 1 Source-Anchored Inversion-Free Forgery

Input: x^w, p^{tar}, proxy $(\mathcal{E}_a, \mathcal{D}_a, V_a)$, discrete times $0 = t_0 < \cdots < t_J \le 1$

$z_0^{\mathrm{src}} \leftarrow \mathcal{E}_a(x^w)$, $u_{t_J} \leftarrow z_0^{\mathrm{src}}$
for $j = J, J-1, \ldots, 1$ **do**
  $\epsilon_j \sim \mathcal{N}(0, I)$
  $z_{t_j}^{\mathrm{src}} \leftarrow (1 - t_j) z_0^{\mathrm{src}} + t_j \epsilon_j$
  $z_{t_j}^{\mathrm{tar}} \leftarrow z_{t_j}^{\mathrm{src}} + (u_{t_j} - z_0^{\mathrm{src}})$
  $v_{t_j}^{\mathrm{src}} \leftarrow V_a(z_{t_j}^{\mathrm{src}}, t_j, p^{\emptyset})$
  $v_{t_j}^{\mathrm{tar}} \leftarrow V_a(z_{t_j}^{\mathrm{tar}}, t_j, p^{\mathrm{tar}})$
  $\Delta v_{t_j} \leftarrow v_{t_j}^{\mathrm{tar}} - v_{t_j}^{\mathrm{src}}$
  $\Delta u_{t_j} \leftarrow (t_{j-1} - t_j)\Delta v_{t_j}$
  $u_{t_{j-1}} \leftarrow u_{t_j} + \Delta u_{t_j}$
end for
$z_0^{\mathrm{tar}} \leftarrow u_0$
return $y \leftarrow \mathcal{D}_a(z_0^{\mathrm{tar}})$

The reconstruction dependency. An inversion-based attack estimates initial noise in the proxy model before target-conditioned regeneration. Under model mismatch, reconstruction uses the proxy's representation and dynamics without enforcing the victim's watermark correspondence. Recall the source latent $z_0^{\mathrm{src}} = \mathcal{E}_a(x^w)$, where $\mathcal{E}_a$ is the proxy encoder. Within a fixed proxy representation, Appendix B.3 decomposes the terminal latents as

$$\begin{aligned} z_0^{\mathrm{inv}} &= z_0^{\mathrm{src}} + r_{\mathrm{inv}} + r_{\mathrm{reg}}, \\ z_0^{\mathrm{tar}} &= z_0^{\mathrm{src}} + r_{\mathrm{sa}}, \end{aligned} \tag{5}$$

where r_{inv} is the empty-prompt reconstruction residual, r_{reg} is the target-conditioned regeneration displacement relative to that reconstruction, and r_{sa} is our accumulated transport displacement.

Effect on victim-side evidence. Let $\Phi_k(x)$ denote continuous victim-side evidence before discrete verification, and let J_ϕ be the Jacobian of $\phi_k(z) = \Phi_k(\mathcal{D}_a(z))$ at z_0^{src}. Under local smoothness assumptions, Appendix B.2 gives the sufficient retention condition for retaining source-watermark acceptance

$$\delta_{\mathrm{rt}} + \|J_\phi h\| + \frac{L_\phi}{2}\|h\|^2 < \mu_k, \tag{6}$$

where h is the terminal displacement from z_0^{src}, δ_{rt} is the encode–decode evidence-shift magnitude, μ_k is the source's acceptance margin, and L_ϕ bounds local Jacobian variation. For inversion, $h = r_{\mathrm{inv}} + r_{\mathrm{reg}}$ adds the first-order residual response $J_\phi r_{\mathrm{inv}}$. For a fixed conditional displacement, this response increases the magnitude of the linearized evidence shift when it is nonzero and nonnegatively aligned with the remaining response, including the round-trip shift (Appendix B.3).

Source-anchored advantage. Our path avoids initial-noise reconstruction and its associated residual. For a fixed conditional displacement, this removes nonnegative residual terms from a conservative evidence-drift bound (Appendix B.3); it does not imply a universal ordering of realized

attack outcomes. Section 4.5 provides complementary evidence: under the evaluated configurations, cross-family inversion exhibits larger victim-side recovered-latent drift and BitAcc losses than model-matched inversion, whereas our attack maintains low latent drift and BitAcc loss on both evaluated victims.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Data, source generators, and watermarks. We evaluate our proposed attack using source–target prompt pairs from PIE-Bench (Ju et al., 2024). We randomly select ten pairs from each of its ten edit categories, yielding 100 pairs. For each source prompt, a victim diffusion model generates a watermarked image. Our main evaluation uses 256-bit Gaussian Shading (GS) (Yang et al., 2024) with five source generators: Stable Diffusion 1.5 (SD1.5), Stable Diffusion 2.1-base (SD2.1) (Rombach et al., 2022), SDXL-base 1.0 (SDXL) (Podell et al., 2024), Stable Diffusion 3 Medium (SD3-M) (Esser et al., 2024), and FLUX.1-dev (FLUX) (Black Forest Labs, 2024). All images have resolution 512×512. Within each victim setting, all attacks receive the same watermarked source image and prescribed target prompt. Appendix C provides further details on the dataset, source-image generation, and watermark verification.

Attack configurations. Our attack uses FLUX.1-dev as its backbone for all victim models, with source/target guidance scales of 1.5/5.5 and an empty source prompt. From a 28-step sampling schedule, we use the final $J = 24$ steps for the paired updates in Algorithm 1. Our main baseline, SD2.1-Inv, is our implementation of Reprompting (Müller et al., 2025): it encodes the source image, performs DDIM inversion with an empty prompt, and regenerates from the recovered latent under the target prompt. FLUX-Inv applies the same inversion–regeneration attack with FLUX.1-dev as the proxy model. Each attack produces one output per input pair, without detector queries or candidate selection.

Evaluation metrics. We measure *watermark performance* using bit accuracy (BitAcc), detection (Det), and traceability (Trace). BitAcc is the fraction of recovered bits matching the embedded watermark. Det measures whether the watermark verifier accepts the known source watermark. Trace uses the stricter source-watermark threshold calibrated under an assumed population of 10^6 registered watermarks. Both use the verifier's nominal $\mathrm{FPR} = 10^{-6}$ setting. Higher BitAcc, Det, and Trace favor the attacker, as tampered images retain the source attribution. Appendix C.3 gives the thresholds and calculation details. We assess *source-image fidelity* using LPIPS (Zhang et al., 2018), CLIP-I, and DINO (Caron et al., 2021), and *target-prompt alignment* using CLIP-T (Radford et al., 2021).

4.2 CROSS-MODEL EFFECTIVENESS

We evaluate cross-model attacks on GS using the balanced 100-pair prompt sets. In addition to SD2.1-Inv, we instantiate the inversion–regeneration protocol with FLUX.1-dev and its flow-matching inversion path, denoted FLUX-Inv. Table 1 compares all three attacks across five victim generators.

Watermark attack effectiveness. Our method achieves the highest BitAcc, Det, and Trace on all five victims, with Det and Trace of 0.99–1.00. Relative to the source images, its average Det and Trace each decline by only 0.2%, compared with 0.168/0.264 for SD2.1-Inv and 0.080/0.162 for FLUX-Inv. The difference is particularly large for SDXL: our Trace remains at 1.00, whereas the inversion baselines fall to 0.60 and 0.47. On the FLUX victim, SD2.1-Inv drops to 0.684 BitAcc and 0.43 Trace, while our method retains 0.999 and 1.00. Inversion remains effective in favorable model pairings, such as SD2.1-Inv on SD2.1 sources with 0.97 Trace and FLUX-Inv on FLUX sources with 0.99 Trace, but our method preserves watermark evidence consistently across the evaluated generators.

Generation fidelity. Our method achieves the lowest LPIPS and highest CLIP-I and DINO for every victim. Its average LPIPS is 0.1978, compared with 0.4539 for SD2.1-Inv and 0.5730 for FLUX-Inv. On FLUX sources, using the same attacker backbone as FLUX-Inv, it reduces LPIPS from 0.3793 to 0.1105 and increases CLIP-I from 0.9080 to 0.9702, demonstrating stronger *source*

Table 1: **Cross-model forgery of Gaussian Shading.** Attacks share source images within each victim. Higher watermark scores favor the attacker. Parentheses give attack-minus-source changes as decimal fractions (negative denotes loss), approximated from rounded report values. Bold marks the best attack per victim and metric.

		Watermark evidence			Source fidelity			Alignment
Victim	Attack	BitAcc↑	Det↑	Trace↑	LPIPS↓	CLIP-I↑	DINO↑	CLIP-T↑
SD1.5	SD2.1-Inv	0.983 (−0.017)	0.990 (−0.010)	0.990 (−0.010)	0.4478	0.8626	0.7216	32.26
	FLUX-Inv	0.875 (−0.125)	0.980 (−0.020)	0.950 (−0.050)	0.6759	0.8067	0.6171	**32.66**
	Ours	**0.999** (−0.001)	**1.000** (0.000)	**1.000** (0.000)	**0.2603**	**0.9202**	**0.8675**	30.98
SD2.1	SD2.1-Inv	0.972 (−0.028)	0.970 (−0.030)	0.970 (−0.030)	0.3872	0.8865	0.7678	32.13
	FLUX-Inv	0.868 (−0.132)	0.950 (−0.050)	0.890 (−0.110)	0.6348	0.8358	0.6554	**32.45**
	Ours	**0.998** (−0.002)	**1.000** (0.000)	**1.000** (0.000)	**0.2531**	**0.9275**	**0.8594**	31.13
SDXL	SD2.1-Inv	0.734 (−0.254)	0.750 (−0.250)	0.600 (−0.400)	0.5828	0.8006	0.6175	**31.92**
	FLUX-Inv	0.710 (−0.278)	0.770 (−0.230)	0.470 (−0.530)	0.6054	0.8077	0.6333	30.75
	Ours	**0.951** (−0.037)	**1.000** (0.000)	**1.000** (0.000)	**0.1753**	**0.9167**	**0.8743**	29.07
SD3-M	SD2.1-Inv	0.786 (−0.210)	0.820 (−0.180)	0.690 (−0.310)	0.4206	0.8711	0.7381	**32.28**
	FLUX-Inv	0.906 (−0.090)	0.910 (−0.090)	0.890 (−0.110)	0.5698	0.8586	0.7238	32.16
	Ours	**0.983** (−0.013)	**0.990** (−0.010)	**0.990** (−0.010)	**0.1900**	**0.9518**	**0.9118**	30.59
FLUX	SD2.1-Inv	0.684 (−0.316)	0.630 (−0.370)	0.430 (−0.570)	0.4312	0.8598	0.7183	**32.36**
	FLUX-Inv	0.994 (−0.006)	0.990 (−0.010)	0.990 (−0.010)	0.3793	0.9080	0.8383	32.10
	Ours	**0.999** (−0.001)	**1.000** (0.000)	**1.000** (0.000)	**0.1105**	**0.9702**	**0.9500**	30.77
Avg.	SD2.1-Inv	0.832 (−0.165)	0.832 (−0.168)	0.736 (−0.264)	0.4539	0.8561	0.7127	**32.19**
	FLUX-Inv	0.870 (−0.127)	0.920 (−0.080)	0.838 (−0.162)	0.5730	0.8434	0.6936	32.02
	Ours	**0.986** (−0.011)	**0.998** (−0.002)	**0.998** (−0.002)	**0.1978**	**0.9373**	**0.8926**	30.51

Table 2: **Attacks against SemBind-protected GS.** Source columns report unattacked images; other metrics describe attacked outputs. Hash agree. measures source–output semantic-hash agreement. Bold marks the better attack per metric. Higher watermark scores favor the attacker.

		Source watermark		Attack watermark			Source fidelity			Alignment
Victim	Attack	BitAcc	Det	BitAcc↑	Det↑	Hash agree.↑	LPIPS↓	CLIP-I↑	DINO↑	CLIP-T↑
SD2.1	SD2.1-Inv	0.986	1.000	0.913	0.920	0.838	0.388	0.887	0.766	**32.19**
	Ours	0.986	1.000	**0.950**	**0.990**	**0.896**	**0.235**	**0.934**	**0.867**	31.04
FLUX	SD2.1-Inv	0.989	1.000	0.625	0.390	0.821	0.402	0.869	0.736	**32.23**
	Ours	0.989	1.000	**0.972**	**1.000**	**0.930**	**0.105**	**0.972**	**0.951**	30.58
Avg.	SD2.1-Inv	0.988	1.000	0.769	0.655	0.830	0.395	0.878	0.751	**32.21**
	Ours	0.988	1.000	**0.961**	**0.995**	**0.913**	**0.170**	**0.953**	**0.909**	30.81

fidelity. For *target alignment*, our mean CLIP-T is 30.51, below 32.19 and 32.02 for the inversion baselines; four of five victim averages exceed 30, with SDXL at 29.07. Together with the attack output images in Figure 5, these scores indicate compatibility with the target descriptions, although 30 is not an edit-completion threshold. A plausible explanation for the baselines' higher CLIP-T is that full inversion followed by target-conditioned regeneration allows more extensive re-synthesis, while our explicit source anchor better preserves appearance and layout.

4.3 Attacking a Watermark Defense

SemBind (Zhang et al., 2026) binds a diffusion watermark to an image-dependent semantic hash to resist forgery attacks. We evaluate SemBind-protected Gaussian Shading (GS-S) using the same balanced 100 prompt pairs. Table 2 reports results with SD2.1 and FLUX as victim generators, whose sources all pass detection. Protocol and analysis details appear in Appendix D.1.

Attack performance against SemBind. Table 2 shows that our method retains near-complete watermark evidence on the reported SD2.1 and FLUX victims, with average BitAcc 0.961 and Det



Figure 3: Visual comparison of forgery attacks under GS-S. Each panel compares the source, SD2.1-Inv, and ours for one of four edit types. Source-target prompts appear above the images; BitAcc appears below.

0.995, while also improving LPIPS, CLIP-I, and DINO over SD2.1-Inv. The largest gap appears on FLUX, where the inversion baseline changes model family and loses much of the decoded watermark. Lower CLIP-T indicates a trade-off with target alignment.

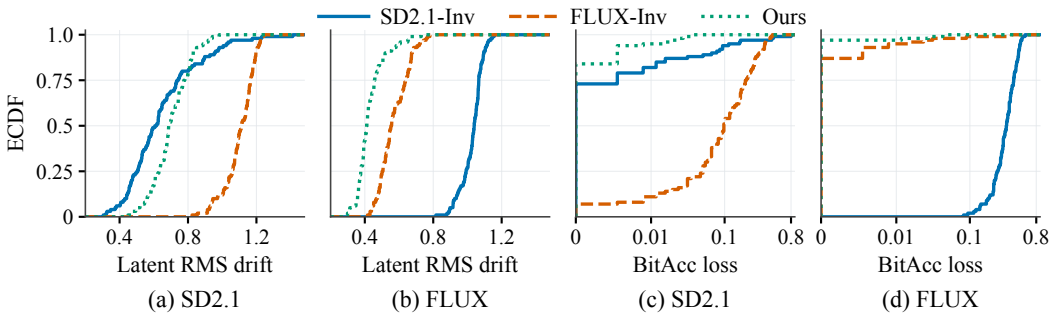
Exposing a defense weakness. Table 2 shows a strong positive association between source–output hash agreement (see Appendix D.1) and watermark retention (BitAcc) across the evaluated attack settings. This is the relevant failure mode for a global semantic binding under local edits: when the target prompt changes an attribute, object, or background while preserving much of the scene, the recomputed hash can remain close enough for watermark evidence to survive. The model-matched SD2.1-Inv row is a useful stress case: even without our source-anchored attack, it retains Det 0.920 on SemBind-protected SD2.1 sources. Appendix D.2 compares this PIE-Bench result with COCO→I2P (Lin et al., 2014; Schramowski et al., 2023) reprompting, which retains much less watermark evidence.

4.4 Qualitative Results

Figure 3 provides a visual comparison of forgery attacks against GS-S across four edit types, using the same source image and prompt pair for both attacks. Our method better preserves source appearance and composition, whereas SD2.1-Inv often changes unrelated details. The object-deletion and recoloring examples illustrate targeted changes with limited disruption to the surrounding scene. Our outputs remain above the watermark detection and traceability thresholds, showing that source watermark evidence can survive semantic manipulation even under SemBind. This highlights the gap between watermark acceptance and semantic integrity. Appendix D.4 provides further visual examples and analysis.

4.5 Victim-Side Evidence Drift

We reuse the SD2.1 and FLUX sources from Table 1 (100 per victim) and their outputs from all three attacks. The victim verifier recovers initial-noise latents $\hat{z}_T^w$ and $\hat{z}_T^y$ from the source and attacked image, respectively. We compare their root-mean-square (RMS) drift and paired BitAcc loss. Appendix D.5 gives the further post-attack diagnostic protocol and metric definitions.



(a) SD2.1 (b) FLUX (c) SD2.1 (d) FLUX

Figure 4: Victim-side latent drift and BitAcc loss. Empirical cumulative distribution functions (ECDFs) of source–output RMS drift (a–b) and BitAcc loss (c–d) over 100 paired samples per victim and attack. Loss axes are linear up to 0.005 and logarithmic thereafter.

Figure 4 shows larger recovered-latent drift and more severe BitAcc losses under cross-family inversion. Ours preserves the source BitAcc on 84/100 SD2.1 and 97/100 FLUX samples. On SD2.1, however, Ours has greater mean RMS drift than SD2.1-Inv (0.702 versus 0.645), yet much lower mean BitAcc loss (0.0017 versus 0.0275). Displacement magnitude alone therefore does not explain watermark retention. The samplewise analysis in Appendix D.5 further shows that larger drift tends to accompany greater BitAcc loss in the high-drift regime, particularly for SD2.1. These observations support more stable evidence retention under our attack, consistent with the source-anchored mechanism in Section 3.3.

5 Conclusion and Discussion

Our study identifies model-dependent reconstruction as a practical bottleneck for forgery attacks on semantic watermarks. The proposed source-anchored, inversion-free forgery attack mitigates this bottleneck, retaining more watermark evidence than the evaluated inversion baselines across multiple victim generators. Experiments on semantic watermarks and SemBind-protected semantic watermarks further show that malicious semantic manipulation can produce outputs that remain accepted by the watermark verifier. These findings demonstrate that forgery attacks need not rely on successful inversion and expose a vulnerability that persists even when watermarks are explicitly bound to image semantics.

The broader security implication is that watermark persistence and semantic integrity can diverge. A manipulated image may still pass verification against the source watermark, but this acceptance alone does not establish that the generator or user associated with that watermark produced the modified content. In the evaluated SemBind settings, our results suggest that verifier acceptance can persist after certain object- or attribute-level changes. Evaluating resistance to such forgery attacks therefore requires assessing verifier acceptance jointly with the semantic changes actually achieved, particularly when attacks differ in target-prompt alignment. Future defenses against semantic forgery should aim to combine robustness to watermark-removal attacks with content-sensitive verification capable of detecting semantically consequential edits, including localized modifications.

AI use statement

We used generative AI tools for feedback on experimental design, including suggestions for additional control experiments that the authors independently assessed. Additional uses included restructuring portions of the manuscript, editing it for readability, and adapting it to the ICLR template. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work examines a false-attribution risk in semantic watermarks: a tampered image may retain a provenance signal even when its content no longer matches the attributed generation event. The purpose of this study is to support defensive evaluation of provenance systems, and watermark presence alone should not be interpreted as proof of semantic integrity or legal responsibility.

REPRODUCIBILITY STATEMENT

Section 3 and Algorithm 1 specify the attack and its information constraints. Appendix A derives the source-relative transport conventions; Appendix B separates source creation, forgery, and verification and states the local sufficient condition and controlled reconstruction-residual comparison; Appendix C records the experimental settings, sample pairing, and watermark-verification conventions. Appendix D provides GS-S configuration details, results for additional watermarking methods, and qualitative evidence. These materials specify the method, analytical assumptions, and evaluation protocol to support independent reproduction and assessment of the reported results.

REFERENCES

Ali Al-Haj. Combined DWT-DCT digital image watermarking. *Journal of Computer Science*, 3(9): 740–746, Sep 2007. doi: 10.3844/jcssp.2007.740.746.

Charlotte Bird, Eddie Ungless, and Atoosa Kasirzadeh. Typology of risks of generative text-to-image models. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '23, pp. 396–410, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702310. doi: 10.1145/3600211.3604722.

Black Forest Labs. FLUX. `https://github.com/black-forest-labs/flux`, 2024.

Tu Bui, Shruti Agarwal, Ning Yu, and John Collomosse. RoSteALS: Robust steganography using autoencoder latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 933–942, June 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2007. ISBN 9780080555805.

Ziping Dong, Chao Shuai, Zhongjie Ba, Peng Cheng, Zhan Qin, Qinglong Wang, and Kui Ren. WMCopier: Forging invisible watermarks on arbitrary images. In *Advances in Neural Information Processing Systems*, volume 38, pp. 153114–153144, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 2024.

Jianwei Fei, Yunshu Dai, Zhihua Xia, Fangjun Huang, and Jiantao Zhou. OmniMark: Efficient and scalable latent diffusion model fingerprinting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16550–16558, 2025. doi: 10.1609/aaai.v39i16.33818.

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.

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

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

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.

Kokil Jaidka, Tsuhan Chen, Simon Chesterman, Wynne Hsu, Min-Yen Kan, Mohan Kankanhalli, Mong Li Lee, Gyula Seres, Terence Sim, Araz Taeihagh, Anthony Tung, Xiaokui Xiao, and Audrey Yue. Misinformation, Disinformation, and Generative AI: Implications for Perception and Policy. *Digit. Gov.: Res. Pract.*, 6(1), February 2025. doi: 10.1145/3689372.

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. PnP inversion: Boosting diffusion-based editing with 3 lines of code. In *The Twelfth International Conference on Learning Representations*, 2024.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. FlowEdit: Inversion-free text-based editing using pre-trained flow models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19721–19730, 2025.

Xiaolong Li, Bin Yang, Daofang Cheng, and Tieyong Zeng. A generalization of LSB matching. *IEEE Signal Processing Letters*, 16(2):69–72, 2009. doi: 10.1109/LSP.2008.2008947.

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

Yisroel Mirsky and Wenke Lee. The creation and detection of deepfakes: A survey. *ACM Comput. Surv.*, 54(1), January 2021. ISSN 0360-0300. doi: 10.1145/3425780.

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

Andreas Müller, Denis Lukovnikov, Jonas Thietke, Asja Fischer, and Erwin Quiring. Black-box forgery attacks on semantic watermarks for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20937–20946, 2025.

K. A. Navas, Mathews Cheriyan Ajay, M. Lekshmi, Tampy S. Archana, and M. Sasikumar. DWT-DCT-SVD based watermarking. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, pp. 271–274, 2008. doi: 10.1109/COMSWA.2008.4554423.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205, October 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 *The Twelfth International Conference on Learning Representations*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, 2021.

Jie Ren, Han Xu, Pengfei He, Yingqian Cui, Shenglai Zeng, Jiankun Zhang, Hongzhi Wen, Jiayuan Ding, Pei Huang, Lingjuan Lyu, Hui Liu, Yi Chang, and Jiliang Tang. Copyright protection in generative AI: A technical perspective. *arXiv preprint arXiv:2402.02333*, 2024.

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*, pp. 118–136, Cham, 2025. Springer Nature Switzerland.

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.

Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22522–22531, 2023.

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 (CVPR)*, pp. 2117–2126, June 2020.

Cristian Vaccari and Andrew Chadwick. Deepfakes and disinformation: Exploring the impact of synthetic political video on deception, uncertainty, and trust in news. *Social Media + Society*, 6 (1):2056305120903408, 2020. doi: 10.1177/2056305120903408.

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

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

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

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. doi: 10.1109/CVPR.2018.00068.

Xin Zhang, Zijin Yang, Kejiang Chen, Linfeng Ma, Weiming Zhang, and Nenghai Yu. SemBind: Binding diffusion watermarks to semantics against black-box forgery attacks. In *International Conference on Machine Learning*, 2026.

Chaoyi Zhu, Zaitang Li, Renyi Yang, Robert Birke, Pin-Yu Chen, Tsung-Yi Ho, and Lydia Y. Chen. Optimization-free universal watermark forgery with regenerative diffusion models. *arXiv preprint arXiv:2506.06018*, 2025. doi: 10.48550/arXiv.2506.06018.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. HiDDeN: Hiding data with deep networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 657–672, September 2018.

Organization. This appendix follows the paper's evidence chain. Appendix A extends the method in Section 3.2 with its time convention and source-relative displacement identity. Appendix B supports the mechanism analysis in Section 3.3 by separating source creation, the forgery operation, and victim-side verification, and by deriving a local sufficient condition together with a controlled comparison of the reconstruction residual. Appendix C then records the prompt-pair protocol, attack settings, and watermark-verification rules. Appendix D provides supporting GS and GS-S analyses, extends the evaluation to PRC, Spherical, PRC-S, and Spherical-S, and presents additional qualitative results. Throughout the appendix, verifier acceptance and automatic prompt-alignment scores are treated as separate measurements rather than proof that a requested semantic edit was completed.

A Method Details for Source-Anchored Transport

This appendix expands the source-anchored transport in Section 3.2. Its purpose is to make the time convention and the accumulated displacement explicit. Experimental settings are recorded separately in Appendix C.2.

A.1 Time and Conditioning Conventions

Let $0 = t_0 < t_1 < \cdots < t_J \leq 1$ denote the selected time points along the proxy model's flow-matching trajectory. Here $t = 0$ corresponds to a clean image latent, while larger t assigns greater weight to Gaussian noise in the interpolated states. The time points need not be uniformly spaced. Only their ordering and the signed increments $t_{j-1} - t_j$ enter the update in Eq. (4).

At step j, the source and target states use the same Gaussian draw ϵ_j. The source branch is conditioned on the empty prompt and the target branch on p^{tar}; the shorthand $V_a(z, t, p)$ absorbs the fixed branch-specific guidance used by the predictor. Sharing noise does not make the two velocity predictions equal. It ensures instead that their state difference contains only the accumulated target displacement:

$$z_{t_j}^{\mathrm{tar}} - z_{t_j}^{\mathrm{src}} = u_{t_j} - z_0^{\mathrm{src}}. \tag{7}$$

The endpoint $u_{t_J} = z_0^{\mathrm{src}} = \mathcal{E}_a(x^w)$ therefore anchors the transport to the observed image in the proxy representation, rather than to an estimated initial-noise latent.

A.2 Accumulated Source-Relative Displacement

Define $d_{t_j} = u_{t_j} - z_0^{\mathrm{src}}$ and

$$\Delta v_{t_j} = V_a(z_{t_j}^{\mathrm{tar}}, t_j, p^{\mathrm{tar}}) - V_a(z_{t_j}^{\mathrm{src}}, t_j, p^{\emptyset}). \tag{8}$$

Subtracting the fixed source anchor from Eq. (4) gives

$$d_{t_{j-1}} = d_{t_j} + (t_{j-1} - t_j)\Delta v_{t_j}, \qquad d_{t_J} = 0. \tag{9}$$

Consequently,

$$z_0^{\mathrm{tar}} - z_0^{\mathrm{src}} = d_0 = \sum_{j=1}^{J} (t_{j-1} - t_j)\Delta v_{t_j}, \tag{10}$$

where each velocity difference is evaluated along the recursively generated states. Although the additive noise cancels from the paired-state difference in Eq. (7), it can still affect both nonlinear velocity predictions and hence the resulting displacement.

Equation (10) is an exact decomposition of the terminal latent into the source anchor and the accumulated velocity-field differences. It explains how source anchoring enters the computation, while the semantic effect and watermark retention of the decoded output are evaluated separately in image space.

B Evidence Path for the Reconstruction Dependency

This appendix formalizes the reconstruction dependency discussed in Section 3.3. We derive a local sufficient condition for watermark retention and analyze how a reconstruction residual affects the evidence response and its conservative bound.

B.1 Analysis Setup

The victim generates x^w from a message-bearing initial-noise latent; the attacker maps (x^w, p^{tar}) to y, whose watermark is assessed by the victim verifier. Fix this verifier and a proxy representation for each analytical comparison. We retain the notation $z_0^{\mathrm{src}} = \mathcal{E}_a(x^w)$ from Section 3.2: $\mathcal{E}_a$ is the encoder, and z_0^{src} is its clean source-image latent, not the victim's initial noise.

Let $\Phi_k(x) \in \mathbb{R}^q$ collect continuous quantities sufficient for the verifier's subsequent discrete decision. For Gaussian Shading, these can be the recovered initial-noise values before bin assignment and message decoding. Define

$$\phi_k(z) = \Phi_k(\mathcal{D}_a(z)), \qquad J_\phi = D\phi_k(z_0^{\mathrm{src}}). \tag{11}$$

Here $D\phi_k$ denotes the Jacobian; J remains the number of transport updates. All vector norms are Euclidean, with the induced norm for Jacobians. The fixed verification rule, including the source identity when applicable, induces an acceptance region $\mathcal{A}_k \subseteq \mathbb{R}^q$. Assume the accepted source lies in its interior, with margin

$$\mu_k = \mathrm{dist}(\Phi_k(x^w), \mathcal{A}_k^{\mathrm{c}}) > 0. \tag{12}$$

The proxy encode–decode round trip contributes

$$e_{\mathrm{rt}} = \phi_k(z_0^{\mathrm{src}}) - \Phi_k(x^w), \qquad \delta_{\mathrm{rt}} = \|e_{\mathrm{rt}}\|. \tag{13}$$

Thus the analysis does not assume exact reconstruction of the source image or its watermark evidence.

B.2 A Local Sufficient Condition

Assume ϕ_k is continuously differentiable on a convex neighborhood $\mathcal{U}$ of z_0^{src}, with an L_ϕ-Lipschitz Jacobian. For a displacement h with $z_0^{\mathrm{src}} + h \in \mathcal{U}$, Taylor expansion gives

$$\phi_k(z_0^{\mathrm{src}} + h) - \Phi_k(x^w) = e_{\mathrm{rt}} + J_\phi h + R(h), \qquad \|R(h)\| \le \frac{L_\phi}{2}\|h\|^2. \tag{14}$$

Proposition 1 (local watermark retention). The output $\mathcal{D}_a(z_0^{\mathrm{src}} + h)$ is accepted whenever

$$U(h) := \delta_{\mathrm{rt}} + \|J_\phi h\| + \frac{L_\phi}{2}\|h\|^2 < \mu_k. \tag{15}$$

Proof. The Taylor remainder is the integral of $[D\phi_k(z_0^{\mathrm{src}} + sh) - J_\phi]h$ over $s \in [0,1]$. Its norm is at most $\int_0^1 L_\phi s\|h\|^2 \,\mathrm{d}s = L_\phi\|h\|^2/2$. The triangle inequality then bounds the source-to-output evidence displacement by $U(h)$. If $U(h) < \mu_k$, the output evidence lies inside the acceptance ball around $\Phi_k(x^w)$. □

This sufficient condition separates the round-trip discrepancy, linear evidence response, and nonlinear remainder. It is an analytical condition; the attack requires no access to the key, verifier derivatives, or margin.

B.3 The Reconstruction Residual in Inversion-Based Forgery

For source anchoring, Eq. (10) gives

$$r_{\mathrm{sa}} = z_0^{\mathrm{tar}} - z_0^{\mathrm{src}} = d_0 = \sum_{j=1}^{J}(t_{j-1} - t_j)\Delta v_{t_j}. \tag{16}$$

For inversion, let $\widehat{z}_T$ be the proxy initial-noise estimate and $F_a(\widehat{z}_T, p)$ the terminal clean latent generated from it under prompt p, with the generation configuration fixed. Define

$$\begin{aligned} r_{\mathrm{inv}} &= F_a(\widehat{z}_T, p^{\emptyset}) - z_0^{\mathrm{src}}, \\ r_{\mathrm{reg}} &= F_a(\widehat{z}_T, p^{\mathrm{tar}}) - F_a(\widehat{z}_T, p^{\emptyset}). \end{aligned} \tag{17}$$

The empty-prompt generation is an analytical reconstruction reference, not an additional attack output. The exact terminal decomposition is

$$z_0^{\mathrm{inv}} = F_a(\widehat{z}_T, p^{\mathrm{tar}}) = z_0^{\mathrm{src}} + r_{\mathrm{inv}} + r_{\mathrm{reg}}. \tag{18}$$

Thus inversion introduces the first-order residual response $J_\phi r_{\mathrm{inv}}$ alongside $J_\phi r_{\mathrm{reg}}$. The reconstruction procedure does not explicitly constrain this response through the victim verifier.

Directional effect. To isolate the residual, fix a conditional displacement d and compare $h = d$ with $h = d + r$, with both endpoints in the same neighborhood $\mathcal{U}$. Set $b(d) = e_{\mathrm{rt}} + J_\phi d$. The squared norms of the linearized evidence displacements satisfy

$$\|b(d) + J_\phi r\|^2 - \|b(d)\|^2 = 2\langle b(d), J_\phi r\rangle + \|J_\phi r\|^2. \tag{19}$$

If $J_\phi r \neq 0$ and $\langle b(d), J_\phi r\rangle \geq 0$, adding the residual strictly increases the linearized evidence displacement. Opposing responses can instead cancel. The residual's effect therefore depends on its direction and the verifier's sensitivity, not its magnitude alone; the finite-displacement response also includes $R(h)$.

Corollary (conservative residual bound). Under the same assumptions, Proposition 1 and the triangle inequality give the conservative bounds

$$U_{\mathrm{sa}}(d) = \delta_{\mathrm{rt}} + \|J_\phi d\| + \frac{L_\phi}{2}\|d\|^2, \tag{20}$$

$$U_{\mathrm{inv}}(d, r) = \delta_{\mathrm{rt}} + \|J_\phi d\| + \|J_\phi r\| + \frac{L_\phi}{2}(\|d\| + \|r\|)^2. \tag{21}$$

Subtracting the first from the second yields

$$U_{\mathrm{inv}}(d, r) - U_{\mathrm{sa}}(d) = \|J_\phi r\| + L_\phi \|d\|\|r\| + \frac{L_\phi}{2}\|r\|^2 \geq 0. \tag{22}$$

Source anchoring avoids the reconstruction stage and its explicit residual terms in this bound. This controlled comparison holds the proxy and d fixed; it neither assumes that the realized r_{sa} and r_{reg} coincide nor orders their actual evidence displacements.

B.4 Connection to the Reported Evidence

Under model mismatch, proxy reconstruction transfers a victim-generated image through a different representation and dynamics. The analysis identifies how its residual enters watermark verification; it does not establish that mismatch always enlarges the residual. Changing proxies also changes the round-trip discrepancy and evidence map. Section 4.5 measures victim-side recovered-latent drift and BitAcc loss, providing complementary outcome evidence rather than estimates of r_{inv}, J_ϕ, or their alignment. Evidence-space distance alone does not determine discrete BitAcc loss. Semantic manipulation is evaluated separately from watermark retention.

C Experimental Protocol and Watermark Verification

C.1 PIE-Bench Prompt Pairs

PIE-Bench (Ju et al., 2024) provides 700 image-editing examples organized into ten categories, with source and target prompts and editing annotations. For each selected pair, the victim model receives the source prompt p_i^{src} and generates a watermarked source image x_i^w. A forgery attack then receives only $(x_i^w, p_i^{\mathrm{tar}})$ and produces a new image; the source prompt is withheld from the attacker and used only for evaluator-side prompt-pair diagnostics. Our evaluation therefore uses the benchmark's prompt specifications, rather than its original photographs or editing masks.

Table 3: Source-generation and GS verification settings at 512×512. Source generation guidance and victim-side inversion guidance are distinct controls. Latent shapes are unpacked (C, H, W) representations, and GS factors are $(\texttt{ch_factor}, \texttt{hw_factor})$.

Victim	Gen. steps	Gen. guidance	Inv. steps	Latent shape	GS factors
SD1.5	50	7.5	50	$4 \times 64 \times 64$	$(1, 8)$
SD2.1	50	7.5	50	$4 \times 64 \times 64$	$(1, 8)$
SDXL	50	7.5	50	$4 \times 64 \times 64$	$(1, 8)$
SD3-M	28	7.0	28	$16 \times 64 \times 64$	$(1, 16)$
FLUX	20	3.5	20	$16 \times 64 \times 64$	$(1, 16)$

The balanced selection contains ten examples from each category: random/mixed edits (type 0), object replacement (1), object addition (2), object deletion (3), content changes (4), pose changes (5), color changes (6), material changes (7), background changes (8), and style changes (9). We randomly select ten sample IDs per category, yielding 100 prompt pairs. Prompt-annotation brackets are stripped before generation and evaluation.

C.2 Generation, Verification, and Baseline Settings

Table 3 specifies the victim-side source generators and the corresponding latent-recovery settings. Source generation uses the source prompt, whereas the watermark recovery uses an empty prompt and guidance 1.0. SD1.5, SD2.1, and SDXL use DPM-Solver++ for source generation and DDIM inversion. For SD3-M and FLUX, generation follows the native flow-matching Euler schedule, while verification uses reverse Euler integration over the corresponding flow-matching schedule. The GS payload is 256 bits, and the channel/spatial repetition factors are chosen to produce the same payload length in the different latent representations. Sources are generated once for each victim and watermark configuration and reused across attack arms.

Following the Reprompting inversion–regeneration attack protocol (Müller et al., 2025), SD2.1-Inv uses 50 DDIM inversion steps with guidance 1.0 and an empty prompt, followed by 50 target-conditioned generation steps with guidance 7.5. As a complementary instantiation of this inversion-based protocol, FLUX-Inv uses 20 flow-matching inversion steps with guidance 1.0, followed by 20 target-conditioned generation steps with guidance 3.5. Both follow the same attack protocol but differ in the proxy backbone and its associated inversion and generation configurations.

C.3 Watermark Detection and Traceability

Let d_w denote the watermark length, $b_i \in \{0, 1\}^{d_w}$ the binary watermark embedded in source image x_i^w, and $\widehat{b}_i(y)$ the watermark recovered from image y by the victim-side verifier. For GS and GS-S, $d_w = 256$ and

$$B_i(y) = \frac{1}{d_w} \sum_{\ell=1}^{d_w} \mathbf{1}[\widehat{b}_{i,\ell}(y) = b_{i,\ell}], \qquad \mathrm{BitAcc} = \frac{1}{N} \sum_{i=1}^{N} B_i(y_i). \tag{23}$$

Our reported rates are

$$\mathrm{Det} = \frac{1}{N} \sum_{i} \mathbf{1}[B_i(y_i) \geq \tau_{\mathrm{det}}], \qquad \mathrm{Trace} = \frac{1}{N} \sum_{i} \mathbf{1}[B_i(y_i) \geq \tau_{\mathrm{trace}}]. \tag{24}$$

For GS, $\tau_{\mathrm{det}} = 166/256$ and $\tau_{\mathrm{trace}} = 183/256$. Det is the acceptance rate for the known source watermark, whereas Trace applies the stricter threshold calibrated under an assumed population of $U = 10^6$ registered watermarks.

Nominal false-positive calibration and equality convention. Under an idealized null of independent equiprobable recovered bits, $X \sim \mathrm{Binomial}(d_w, 1/2)$. For candidate thresholds in

$\mathcal{T}_{d_w} = \{0, 1/d_w, \ldots, 1\}$, the released GS verifier (Yang et al., 2024) uses

$$\begin{aligned} q_{d_w}(\tau) &= \Pr[X/d_w > \tau] = \sum_{j=d_w\tau+1}^{d_w} \binom{d_w}{j} 2^{-d_w}, \\ \tau_{\text{det}} &= \min\{\tau \in \mathcal{T}_{d_w} : q_{d_w}(\tau) \le \alpha\}, \\ \tau_{\text{trace}} &= \min\{\tau \in \mathcal{T}_{d_w} : U q_{d_w}(\tau) \le \alpha\}, \end{aligned} \tag{25}$$

with $\alpha = 10^{-6}$ and $U = 10^6$. Thus, the verifier selects each threshold using a strict upper tail but applies it with the inclusive decision rule $B_i \ge \tau$. The probability mass at equality is excluded during calibration but included during acceptance. Moreover, $U q_{d_w}(\tau)$ is the union-bound multiuser calibration used by GS. Under independent user tests, the exact family-wise probability would be $1 - (1 - q_{d_w}(\tau))^U$. We follow the released verifier for consistency with the reported experiments and treat α and U as calibration settings rather than empirical guarantees.

D Supplementary Evaluation

D.1 SemBind Configuration

SemBind (Zhang et al., 2026) binds a base semantic watermark to image semantics through an image-dependent binary hash. For GS-S, the semantic masker extracts DINOv2-giant (Oquab et al., 2024) CLS features and maps them to a 1024-bit code. This code determines a keyed sign mask that modulates the GS latent watermark. During verification, the semantic code is recomputed from the image under test, and the corresponding mask is removed before the GS payload is decoded. Verification therefore depends on both the semantic decoding context and the underlying GS watermark signal.

We use binarization parameter 12, mask ratio $\sigma = 1$, and permutation seed 1. The sign mask is tiled to the latent dimensions and permuted before modulation. Apart from this semantic-binding layer, the diffusion-model and base-watermark configurations follow the corresponding GS settings in Appendix C.2.

For a watermarked source x^w and an attacked output y, we report their semantic-hash agreement as

$$H(x^w, y) = \frac{1}{1024} \sum_{j=1}^{1024} \mathbf{1}[h_j(x^w) = h_j(y)]. \tag{26}$$

Here $h_j(x) \in \{0, 1\}$ is the j-th semantic-hash bit. Thus, $H = 1$ denotes identical hashes, whereas intermediate values quantify partial bitwise agreement; $H = 0.5$ is the reference level for independent, balanced binary hashes. Hash agreement measures similarity of the semantic decoding contexts, but does not by itself establish successful watermark recovery.

D.2 SemBind under Cross-Prompt Reprompting and Local Semantic Forgery

We compare GS-S under cross-prompt reprompting and local semantic forgery to examine how SemBind behaves when source and target prompts have different degrees of semantic relatedness. The local-forgery results reuse the SD2.1/SD2.1-Inv entry in Table 2.

Protocol. The victim and inversion-based attacker both use SD2.1, with 100 prompt pairs in each scenario. Local semantic forgery uses the balanced PIE-Bench selection in Appendix C.1, whose target prompts specify localized changes to their paired source prompts. For cross-prompt reprompting, we independently select COCO2017 validation captions (Lin et al., 2014) and I2P prompts (Schramowski et al., 2023) and pair them to form the source–target prompt set. Because the pairing is independent rather than semantically matched, this set represents a low-relatedness cross-prompt setting.

Results and interpretation. Table 4 shows higher watermark retention under local semantic forgery: Det is 0.920, compared with 0.030 under cross-prompt reprompting, accompanied by higher BitAcc and hash agreement. The mean hash agreement for cross-prompt reprompting is

Table 4: SemBind-protected Gaussian Shading under cross-prompt reprompting (COCO→I2P) and local semantic forgery (PIE-Bench).

Setting	BitAcc	Det	Hash agreement
COCO→I2P	0.4762	0.030	0.4894
PIE-Bench	0.913	0.920	0.838

0.489, close to the 0.5 reference level for independent balanced hashes, whereas local semantic forgery reaches 0.838. This difference is consistent with localized edits preserving more of the source semantic context used by SemBind. Within the 100 cross-prompt examples, per-example hash agreement and output BitAcc have Pearson correlation $r = 0.788$, indicating a positive association between preservation of the semantic decoding context and watermark recovery.

D.3 Additional Watermark Results

This subsection extends the evaluation to PRC Watermark (Gunn et al., 2025), Spherical Watermark (Hu et al., 2026), and their SemBind-protected variants. All settings use the balanced PIE-Bench split of 100 prompt pairs, with ten source–target pairs from each edit type. Within each victim setting, the attack arms share the same watermarked source images, prompt pairs, and seeds. Both watermark families use $d_w = 512$-bit watermarks. Using the verification-threshold definitions in Appendix C.3 with $\alpha = 10^{-6}$ and $U = 10^6$ yields $\tau_{\mathrm{det}} = 310/512 = 0.60546875$ and $\tau_{\mathrm{trace}} = 335/512 = 0.654296875$. For Det, PRC and PRC-S are the exceptions to the GS-style evaluation and instead use the native PRC detector. Spherical and Spherical-S follow the GS evaluation in Eq. (24). Across all experiments involving PRC or Spherical Watermark, we use the parameter settings recommended in the corresponding papers and released implementations.

D.3.1 Results on PRC and Spherical Watermarks

Table 5 reports attacks against PRC and Spherical Watermark on SD1.5, SD2.1, SD3-M, and FLUX. Across both watermark families, our method achieves the highest BitAcc, Det, Trace, and source fidelity for every evaluated victim. Averaged over the evaluated victims, ours reaches BitAcc/Det/Trace of $0.924/0.898/0.848$ under PRC and $0.944/0.980/0.970$ under Spherical Watermark.

The inversion baselines reproduce the model-pairing sensitivity observed for GS. Among these baselines, SD2.1-Inv retains substantially more watermark evidence on SD1.5 and SD2.1, whereas FLUX-Inv is substantially stronger on FLUX; watermark evidence deteriorates when the attacker and victim belong to mismatched model families. The baselines obtain higher CLIP-T but substantially lower source fidelity than ours, consistent with more extensive target-conditioned resynthesis. These results show that the retention advantage of our proposed method and the model-dependent inversion bottleneck extend beyond Gaussian Shading to two additional watermark constructions.

D.3.2 Results on PRC-S and Spherical-S

Table 6 reports attacks against SemBind-protected PRC (PRC-S) and Spherical Watermark (Spherical-S) on SD2.1 and FLUX. Each attack cell contains 100 paired outputs and uses a 1024-bit semantic hash. In the attack rows, Hash agree. is the bitwise agreement between the semantic hashes of the attacked output and its watermarked source. It measures how much of the source semantic context remains after tampering.

Our method improves watermark evidence, attack–source Hash agreement, and source fidelity over SD2.1-Inv in every PRC-S and Spherical-S setting. The largest gap occurs on FLUX, where SD2.1-Inv changes model family and yields zero Det under both defenses, whereas ours reaches 0.910 under PRC-S and 0.980 under Spherical-S. Averaged over the two victims, ours raises BitAcc from 0.633 to 0.874 for PRC-S and from 0.663 to 0.903 for Spherical-S. In every defense–victim pair, the attack with higher attack–source Hash agreement also has higher BitAcc, Det, and Trace. This co-variation is consistent with SemBind's mechanism: preserving the source semantic context helps

Table 5: **Attacks against PRC and Spherical Watermark.** Source-average rows and attack *Avg.* rows average equally over the four displayed victims. Higher watermark scores favor the attacker. Bold marks the best attack per victim and metric.

		Watermark evidence			Source fidelity			Alignment
Victim	Attack	BitAcc↑	Det↑	Trace↑	LPIPS↓	CLIP-I↑	DINO↑	CLIP-T↑
PRC Source avg.		0.985	0.978	0.970	–	–	–	–
SD1.5	SD2.1-Inv	0.879	0.850	0.760	0.4455	0.8585	0.7255	32.36
	FLUX-Inv	0.506	0.080	0.020	0.6842	0.8005	0.6085	**32.66**
	Ours	**0.945**	**0.940**	**0.890**	**0.2524**	**0.9175**	**0.8623**	31.05
SD2.1	SD2.1-Inv	0.872	0.780	0.740	0.3760	0.8984	0.7816	32.31
	FLUX-Inv	0.532	0.080	0.060	0.6390	0.8297	0.6576	**32.90**
	Ours	**0.929**	**0.930**	**0.860**	**0.2605**	**0.9212**	**0.8401**	31.58
SD3-M	SD2.1-Inv	0.501	0.000	0.000	0.4076	0.8790	0.7373	31.93
	FLUX-Inv	0.547	0.140	0.090	0.5791	0.8505	0.7177	**32.03**
	Ours	**0.856**	**0.760**	**0.710**	**0.1876**	**0.9506**	**0.9117**	30.83
FLUX	SD2.1-Inv	0.501	0.000	0.000	0.4118	0.8645	0.7337	**32.57**
	FLUX-Inv	0.854	0.810	0.710	0.3597	0.9121	0.8467	31.87
	Ours	**0.966**	**0.960**	**0.930**	**0.1100**	**0.9673**	**0.9382**	30.81
Avg.	SD2.1-Inv	0.688	0.408	0.375	0.4102	0.8751	0.7445	32.29
	FLUX-Inv	0.610	0.278	0.220	0.5655	0.8482	0.7076	**32.36**
	Ours	**0.924**	**0.898**	**0.848**	**0.2026**	**0.9392**	**0.8881**	31.07
Spherical Source avg.		0.987	0.993	0.990	–	–	–	–
SD1.5	SD2.1-Inv	0.910	0.940	0.910	0.4331	0.8725	0.7270	32.03
	FLUX-Inv	0.620	0.460	0.280	0.7014	0.8004	0.5932	**32.44**
	Ours	**0.958**	**1.000**	**1.000**	**0.2525**	**0.9219**	**0.8656**	30.89
SD2.1	SD2.1-Inv	0.891	0.930	0.880	0.3643	0.8940	0.7852	31.90
	FLUX-Inv	0.608	0.410	0.230	0.6558	0.8247	0.6467	**32.34**
	Ours	**0.947**	**0.990**	**0.990**	**0.2395**	**0.9295**	**0.8728**	30.99
SD3-M	SD2.1-Inv	0.538	0.080	0.020	0.4114	0.8716	0.7420	**32.10**
	FLUX-Inv	0.665	0.540	0.450	0.5701	0.8458	0.7253	32.06
	Ours	**0.896**	**0.940**	**0.910**	**0.1829**	**0.9522**	**0.9105**	30.84
FLUX	SD2.1-Inv	0.505	0.000	0.000	0.4154	0.8627	0.7329	**32.28**
	FLUX-Inv	0.907	0.960	0.930	0.3908	0.9009	0.8351	31.95
	Ours	**0.975**	**0.990**	**0.980**	**0.1030**	**0.9703**	**0.9526**	30.51
Avg.	SD2.1-Inv	0.711	0.488	0.453	0.4060	0.8752	0.7468	32.08
	FLUX-Inv	0.700	0.593	0.473	0.5795	0.8430	0.7001	**32.20**
	Ours	**0.944**	**0.980**	**0.970**	**0.1945**	**0.9435**	**0.9004**	30.81

maintain a compatible binding mask, although successful verification also requires the underlying watermark payload to survive.

The model-matched SD2.1-Inv baseline remains accepted on Spherical-S with Det 0.850, showing that the observed weakness is not specific to our attack. Together with the qualitative results in Appendix D.4, these results extend the GS-S finding in the main paper: semantic manipulation can preserve sufficient semantic context and watermark evidence to remain accepted under SemBind across multiple base watermarks. Thus, passing watermark verification does not establish that the attributed image content has remained semantically unchanged.

D.4 Qualitative Results

Figures 5–7 compare Source, SD2.1-Inv, and ours across GS/GS-S, PRC/PRC-S, and Spherical/Spherical-S. Across diverse edits and victims, ours realizes the requested semantic changes while preserving source structure and watermark evidence better than SD2.1-Inv.

Figure 7(e) highlights two conditions required for SemBind verification. SemBind uses an image-derived semantic hash to establish the decoding context, so a high attack–source Hash agreement

Table 6: **Attacks against PRC-S and Spherical-S.** In attack rows, Hash agree. compares the semantic hashes of the attacked output and its watermarked source. Bold marks the better attack per victim and metric.

		Watermark evidence				Source fidelity			Alignment
Victim	Attack	BitAcc↑	Det↑	Trace↑	Hash agree.↑	LPIPS↓	CLIP-I↑	DINO↑	CLIP-T↑
PRC-S Source avg.		0.965	0.965	0.930	–	–	–	–	–
SD2.1	SD2.1-Inv	0.764	0.610	0.530	0.832	0.374	0.882	0.764	**31.79**
	Ours	**0.819**	**0.750**	**0.640**	**0.882**	**0.241**	**0.926**	**0.864**	30.91
FLUX	SD2.1-Inv	0.503	0.000	0.000	0.810	0.421	0.869	0.715	**32.49**
	Ours	**0.929**	**0.910**	**0.860**	**0.930**	**0.105**	**0.968**	**0.940**	30.71
Avg.	SD2.1-Inv	0.633	0.305	0.265	0.821	0.397	0.876	0.739	**32.14**
	Ours	**0.874**	**0.830**	**0.750**	**0.906**	**0.173**	**0.947**	**0.902**	30.81
Spherical-S Source avg.		0.972	0.990	0.985	–	–	–	–	–
SD2.1	SD2.1-Inv	0.818	0.850	0.820	0.832	0.385	0.888	0.773	**32.39**
	Ours	**0.862**	**0.960**	**0.920**	**0.885**	**0.253**	**0.932**	**0.865**	31.22
FLUX	SD2.1-Inv	0.507	0.000	0.000	0.814	0.426	0.856	0.728	**32.48**
	Ours	**0.944**	**0.980**	**0.970**	**0.940**	**0.106**	**0.971**	**0.950**	30.40
Avg.	SD2.1-Inv	0.663	0.425	0.410	0.823	0.406	0.872	0.751	**32.43**
	Ours	**0.903**	**0.970**	**0.945**	**0.912**	**0.179**	**0.951**	**0.908**	30.81

helps preserve source-compatible decoding. However, the watermark evidence required for payload recovery must also survive. SD2.1-Inv achieves the highest Hash agreement (0.911), but its BitAcc drops to 0.479 and detection fails, indicating that inversion preserves the semantic-hash context while disrupting the payload evidence. Ours jointly maintains sufficient Hash agreement (0.802) and recoverable payload evidence, reaching a BitAcc of 0.818 and passing detection. Its advantage therefore comes from preserving both conditions during the target edit, rather than from maximizing Hash agreement alone.

D.5 Victim-Side Latent Drift and Watermark Evidence

This section provides additional experimental details and analyses for the victim-side evidence-drift study in Section 4.5. It reuses the completed GS evaluation in Table 1: 100 paired PIE-Bench sources per victim, with ten pairs from each of the ten edit categories, and outputs from SD2.1-Inv, FLUX-Inv, and Ours.

Victim-side recovery. We recover an initial-noise latent from each source and attacked output using the corresponding victim verifier, an empty prompt, and guidance 1.0. SD2.1 uses 50 recovery steps and FLUX uses 20. Thus, all three attack methods are compared in the same recovery space within each victim. The source reference $\hat{z}_T^w$ is the *recovered* latent from x^w, rather than the true generating noise z_T^w. $\hat{z}_T^y$ denotes the latent recovered from the attacked image y.

Measurements. For a fixed victim, let the recovered latents have shape $h \times w \times c$, with hwc coordinates. Flattening these latents into vectors, define

$$d_{\mathrm{RMS}}(y, x^w; v) = \frac{\|\hat{z}_T^y - \hat{z}_T^w\|_2}{\sqrt{hwc}}, \qquad L_k(y; x^w) = B_k(x^w) - B_k(y). \tag{27}$$

Root-mean-square (RMS) drift measures the aggregate displacement of recovered latent coordinates. Here $B_k(x)$ denotes the BitAcc recovered from image x for watermark key k, so L_k is the signed BitAcc loss from the source to the attacked output. Negative values would indicate improved recovery, although all observed losses in this diagnostic are nonnegative. We report BitAcc and its loss as fractions. The descriptive cutoff $L_k > 0.05$ is independent of the GS detection and traceability thresholds. Latent dimensions and scales differ between victims, so RMS magnitudes should be compared within a victim.

Gaussian Shading (GS)

(a) Random/mixed (type 0)
Source: a **dog** is laying down on a white background
Target: a **lion** is laying down on a white background



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.574 | **Ours** BitAcc=1.000

(b) Random/mixed (type 0)
Source: a painting of a cup with **smoke** coming out of it
Target: a painting of a cup with **a flower** coming out of it



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.863 | **Ours** BitAcc=1.000

(c) Delete object (type 3)
Source: a cartoon of woman is painting a pumpkin **with a cat**
Target: a cartoon of woman is painting a pumpkin



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.512 | **Ours** BitAcc=1.000

(d) Change background (type 8)
Source: a girl in a purple dress holding a bouquet of flowers
Target: a girl in a purple dress holding a bouquet of flowers **on the beach**



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.633 | **Ours** BitAcc=1.000

Gaussian Shading with SemBind (GS-S)

(e) Expression (type 0)
Source: the christmas illustration of a santa's **laughing** face
Target: the christmas illustration of a santa's **angry** face



Source BitAcc=0.977 | SD2.1-Inv BitAcc=0.570 | **Ours** BitAcc=0.832

(f) Change color (type 6)
Source: a **white** mouse with a wand and a star
Target: a **purple** mouse with a wand and a star



Source BitAcc=0.965 | SD2.1-Inv BitAcc=0.488 | **Ours** BitAcc=0.953

(g) Add object (type 2)
Source: a koala is sitting on a tree
Target: a koala **and a bird** is sitting on a tree



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.641 | **Ours** BitAcc=1.000

(h) Change object (type 1)
Source: a cartoon girl sitting on a **backpack**
Target: a cartoon girl sitting on a **rock**



Source BitAcc=0.973 | SD2.1-Inv BitAcc=0.473 | **Ours** BitAcc=0.844

Figure 5: **Qualitative comparisons under GS and GS-S.** The first two rows show Gaussian Shading (GS), and the last two rows show Gaussian Shading with SemBind (GS-S). Changed prompt content is bolded, and detector-side BitAcc is reported below each image.

PRC Watermark (PRC)

(a) Random/mixed (type 0)
Source: **dogs** running in the grass
Target: **rabbits** running in the grass



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.523 | **Ours** BitAcc=1.000

(b) Change object (type 1)
Source: a white brick wall and a **potted plant**
Target: a white brick wall and a **sitted man**



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.496 | **Ours** BitAcc=1.000

(c) Delete object (type 3)
Source: a girl **and a dog** sitting in the forest
Target: a girl sitting in the forest



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.488 | **Ours** BitAcc=1.000

(d) Change content (type 4)
Source: a **round** shape cake with orange frosting on a wooden plate
Target: a **star** shape cake with orange frosting on a wooden plate



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.514 | **Ours** BitAcc=1.000

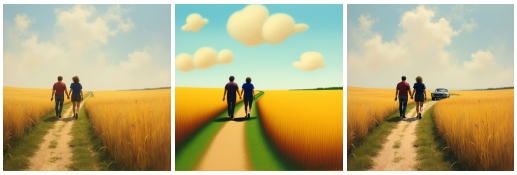
PRC Watermark with SemBind (PRC-S)

(e) Change object (type 1)
Source: a futuristic city with **cars** and people walking in the street
Target: a futuristic city with **rocket** and people walking in the street



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.502 | **Ours** BitAcc=1.000

(f) Add object (type 2)
Source: a painting of two people walking down a path in a field
Target: a painting of two people walking down a path **following a car** in a field



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.477 | **Ours** BitAcc=1.000

(g) Random/mixed (type 0)
Source: a painting of a cup with **smoke** coming out of it
Target: a painting of a cup with **a flower** coming out of it



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.512 | **Ours** BitAcc=1.000

(h) Random/mixed (type 0)
Source: a woman in red sitting on a chair with **castle** in the background
Target: a woman in red sitting on a chair with **ghost** in the background



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.502 | **Ours** BitAcc=1.000

Figure 6: **Qualitative comparisons under PRC and PRC-S.** The first two rows show PRC Watermark (PRC), and the last two rows show PRC Watermark with SemBind (PRC-S). Changed prompt content is bolded, and detector-side BitAcc is reported below each image.

Spherical Watermark (Spherical)

(a) Random/mixed (type 0)
Source: a painting of a cup with **smoke** coming out of it
Target: a painting of a cup with **a flower** coming out of it



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.510 | **Ours** BitAcc=1.000

(b) Change object (type 1)
Source: a futuristic city with **cars** and people walking in the street
Target: a futuristic city with **rocket** and people walking in the street



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.494 | **Ours** BitAcc=0.998

(c) Add object (type 2)
Source: a large brown teddy bear sitting on a white couch
Target: a large brown teddy bear **and a girl** sitting on a white couch



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.502 | **Ours** BitAcc=1.000

(d) Delete object (type 3)
Source: a cartoon of woman is painting a pumpkin **with a cat**
Target: a cartoon of woman is painting a pumpkin



Source BitAcc=0.998 | SD2.1-Inv BitAcc=0.482 | **Ours** BitAcc=0.963

Spherical Watermark with SemBind (Spherical-S)

(e) Change color (type 6)
Source: a watercolor painting of a **white** bunny in a cup of flowers
Target: a watercolor painting of a **pink** bunny in a cup of flowers



Source BitAcc=0.775 | SD2.1-Inv BitAcc=0.479 | **Ours** BitAcc=0.818

(f) Change content (type 4)
Source: a **white** car driving down a road with blurred buildings in the background
Target: a **stripe** car driving down a road with blurred buildings in the background



Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.824 | **Ours** BitAcc=0.961

(g) Add object (type 2)
Source: a bottle of champagne and a hat on a wooden table
Target: a bottle of champagne and a hat **with flowers** on a wooden table



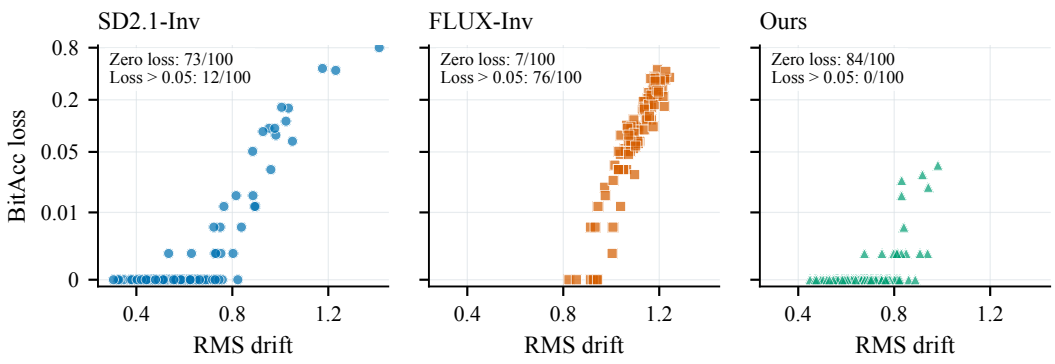
Source BitAcc=0.994 | SD2.1-Inv BitAcc=0.504 | **Ours** BitAcc=0.930

(h) Change object (type 1)
Source: an old man playing an electric **guitar** on a street
Target: an old man playing an electric **piano** on a street

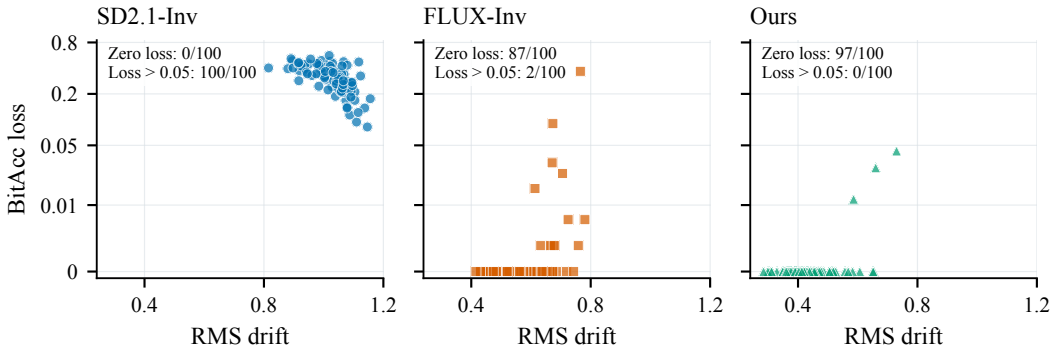


Source BitAcc=1.000 | SD2.1-Inv BitAcc=0.992 | **Ours** BitAcc=0.926

Figure 7: **Qualitative comparisons under Spherical and Spherical-S.** The first two rows show Spherical Watermark, and the last two rows show Spherical Watermark with SemBind (Spherical-S). Changed prompt content is bolded, and detector-side BitAcc is reported below each image.



(a) SD2.1 victim



(b) FLUX victim

Figure 8: **Samplewise latent drift and watermark-evidence loss.** Each panel contains all 100 observations for one victim and attack, with unjittered coordinates. BitAcc loss uses a linear vertical scale up to 0.01 and a logarithmic scale above it. Counts at zero and above 0.05 describe the loss distribution, not detection outcomes.

Distributions and samplewise relationships. The samplewise distributions complement the aggregate results in Section 4.5. Matched inversion often preserves the watermark, with 73/100 and 87/100 zero-loss samples on SD2.1 and FLUX, respectively. Cross-family inversion produces substantially heavier BitAcc-loss tails, whereas Ours has no samples with loss above 0.05 for either victim.

Figure 8 indicates a regime-dependent rather than globally monotone relationship. Once RMS drift exceeds approximately 0.6, larger drift tends to coincide with greater BitAcc loss, and this association is more evident for the SD2.1 victim. At still larger drift, the corresponding samples exhibit severe evidence loss: the original watermark can no longer be reliably recovered by the victim-side verifier.