RETHINKING IMAGE WATERMARKING IN THE ERA OF FRONTIER GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Image watermarks are increasingly used to establish the provenance of AI-generated content. We demonstrate a watermark-removal attack that repeatedly regenerates a watermarked image with GPT-Image-2. The attack requires only API access, without knowledge of the watermarking method, detector access, or local model deployment. It achieves 89% average removal across 13 post-hoc and in-generation methods, with a removal–fidelity trade-off comparable to leading specialized attacks. The evaluated content-based watermarks resist regeneration. However, we show that an existing method that inserts visual concepts from a fixed private inventory is vulnerable to concept extraction. Our extraction attack recovers 79% of its concepts from 3,000 watermarked images. The recovered candidates enable 93.9% removal after ten targeted edits, at a cost to image fidelity, and 70.4% spoofing success when generating images outside the watermarking system. We propose K-SemMark-R, a registered content-based watermark that independently samples prompt-compatible carrier values for each image and stores verified carriers in a private registry. Experiments show that most initially detected images remain detectable by carrier verification after regeneration, although the method does not enroll every prompt. These findings highlight the potential of content-based watermarking and motivate research into image-specific designs that resist regeneration without relying on a shared concept inventory.

1 INTRODUCTION

Recent advances in image generation have made it increasingly difficult to determine an image's origin from its appearance alone. Image watermarking addresses this problem by embedding detectable information into images, supporting the identification of AI-generated content, source attribution, and ownership claims (Gowal et al., 2025). These applications require watermarks to remain detectable after both common image transformations and deliberate removal attempts. Alongside efforts to develop robust watermarking methods (Bui et al., 2023; Lu et al., 2025; Zhang et al., 2024; Gowal et al., 2025; Fernandez et al., 2023; Wen et al., 2023), a growing body of work investigates their vulnerabilities through attacks that evade detection while preserving the image's visual content. These attacks use adversarial optimization (Kassis & Hengartner, 2025) and image regeneration (Liu et al., 2025; Shamshad et al., 2026) to expose vulnerabilities in existing watermarks, motivating reassessment of their robustness as image generation and editing models improve.

In this paper, we revisit the robustness of image watermarking against regeneration with frontier image models, focusing on GPT-Image-2 (OpenAI, 2026c). We show that an attacker can remove watermarks using only API access to this text-and-image-to-image (TI2I) model and a fixed prompt, without knowledge of the watermarking method or access to its detector. Four regeneration rounds achieve an average removal rate of 89% across 13 post-hoc (Lu et al., 2025; Zhang et al., 2024) and in-generation (Wen et al., 2023; Yang et al., 2024) watermarking methods, with a removal-fidelity trade-off comparable to leading specialized attacks. Our results demonstrate that effective watermark removal is possible through a publicly accessible TI2I model, without local model deployment, training, or per-image optimization.

Our evaluation also identifies content-based watermarking as a promising direction: IConMark (Sadasivan et al., 2025) and its hybrid remain resistant to the evaluated removal attacks, including our regeneration attack. IConMark embeds recognizable visual concepts into generated

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.

images, which helps explain its resistance to attacks that preserve image content. However, its reliance on a fixed concept database introduces a different vulnerability. Repeatedly inserting concepts from this limited set allows an adversary to identify recurring objects that are unrelated to the user's prompt. We develop a concept-extraction attack that recovers 79 of IConMark's 100 database concepts from 3,000 watermarked images, without access to the database or watermark detector. Using the extracted candidates, we achieve 93.9% watermark removal after ten targeted edits, with a corresponding loss of image fidelity. Requesting ten extracted concepts also enables spoofing (Saberi et al., 2024; Müller et al., 2025): images generated outside the watermarking system are detected as watermarked in 70.4% of attempts.

Finally, we introduce K-SemMark-R, an image-specific content-based watermark designed to address these limitations. Instead of inserting concepts from a fixed global inventory, it samples values for prompt-compatible scene attributes and stores the verified carriers in a private registry. The method abstains when a prompt provides too few suitable carriers, rather than introducing unrelated content to meet a watermarking requirement. Our measurements show limited recurrence of carrier values, supporting resistance to the reusable-dictionary mechanism exploited against IConMark. On 100 evaluation prompts, the method enrolls 75 images, with 90.7% clean carrier detection among enrolled images. Four rounds of GPT-Image-2 regeneration evade carrier verification on 14.7% of initially detected images. These results show that image-specific semantic carriers can retain detectability under regeneration without relying on a shared concept inventory, offering an alternative design for content-based watermarking.

2 Related Work

2.1 Image Watermarking

Post-hoc methods embed watermarks into existing images. DWT-DCT-SVD modifies transform-domain coefficients (Navas et al., 2008), while RivaGAN (Zhang et al., 2019), StegaStamp (Tancik et al., 2020), and TrustMark (Bui et al., 2023) use learned encoders and decoders to embed and recover messages. SSL Watermark optimizes images in a self-supervised feature space (Fernandez et al., 2022). More recent approaches incorporate generative models: ZoDiac optimizes diffusion latents to watermark existing images (Zhang et al., 2024), and VINE uses generative priors to improve robustness to image editing (Lu et al., 2025). SynthID-Image uses a post-hoc encoder–decoder framework for watermarking at scale (Gowal et al., 2025).

In-generation methods embed invisible signals during image generation. Stable Signature fine-tunes the latent decoder to produce watermarked images (Fernandez et al., 2023). Tree-Ring (Wen et al., 2023), Gaussian Shading (Yang et al., 2024), and PRC (Gunn et al., 2025) encode watermarks in initial noise using Fourier patterns, distribution-preserving encoding, and pseudorandom codes, respectively. ROBIN embeds watermarks in intermediate diffusion states with adversarial optimization (Huang et al., 2024), while SEAL derives an invisible noise watermark from semantic features (Arabi et al., 2025). Content-based watermarking instead uses human-interpretable image content. IConMark (Sadasivan et al., 2025) incorporates concepts from a private database into generated images and detects them using a vision-language model. It also combines these concepts with invisible watermarks such as TrustMark.

2.2 Attacks on Image Watermarks

Watermark-removal attacks seek to evade detection while preserving image fidelity. Beyond standard transformations such as compression, blurring, and cropping, regeneration attacks reconstruct images to suppress watermark signals. Regen-Diff and Regen-VAE use diffusion reconstruction and learned compression (Zhao et al., 2024); Rinse applies repeated diffusion regeneration passes (An et al., 2024); and CtrlRegen reconstructs images from clean noise using image-derived conditioning (Liu et al., 2025). Other approaches include detector-free spectral optimization in UnMarker (Kassis & Hengartner, 2025), reconstruction with an untrained neural network in Deep Image Prior (Liang et al., 2025), and novel-view synthesis in RAVEN (Shamshad et al., 2026). WAVES (An et al., 2024) and W-Bench (Lu et al., 2025) provide broader evaluations of watermark robustness under distortions, regeneration, and editing. Closely related to our approach, Pleimling et al. (2026) use prompted image-to-image models to remove protective perturbations intended to

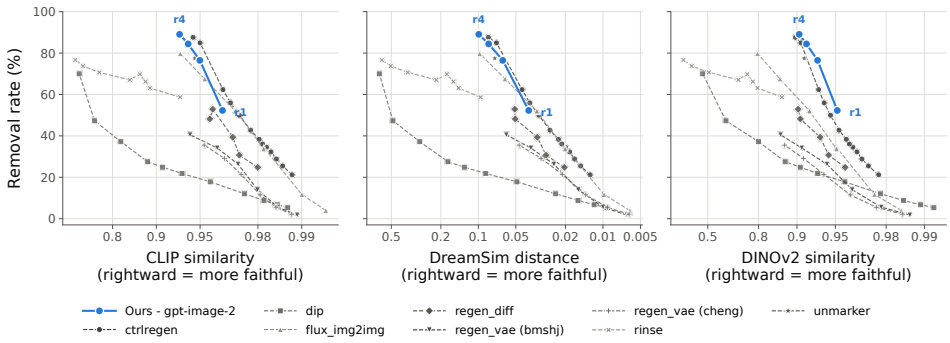

Figure 1: Removal-fidelity trade-offs averaged over 13 non-content-based watermarking methods. Blue points show our GPT-Image-2 regeneration attack over four rounds; gray points show baseline attacks at different strengths. Higher and farther right indicate better removal and fidelity.

prevent unauthorized image use. We study prompted regeneration for provenance watermark removal across watermark families and general-purpose image models, using a fixed prompt without detector feedback.

3 Watermark Removal via Image Regeneration

Recent advances in text-and-image-to-image (TI2I) models have made image editing more accessible, but these capabilities may also facilitate watermark removal. As reviewed in Section 2, existing removal attacks may require specialized knowledge or local computational resources. We investigate whether GPT-Image-2 can achieve watermark removal comparable to specialized attacks using only API access and a simple prompt. To this end, we evaluate a regeneration-based attack that repeatedly re-renders an image while aiming to preserve its visual content.

3.1 Threat Model and Methodology

Threat model. The adversary aims to evade watermark detection while preserving the image's semantic content and perceptual fidelity, without requiring pixel-wise reconstruction. We consider a black-box setting in which the adversary has no knowledge of the image's source or the watermarking method and no access to the watermark detector. The attack requires only the input image and API access to a TI2I model.

Methodology. Let M denote a TI2I model that accepts a text prompt and an input image. Starting from a watermarked image $I_0 = I^*$, the adversary repeatedly applies

$$I_i = M(p^*, I_{i-1}), \qquad i = 1, \ldots, r, \tag{1}$$

where p^* is a fixed regeneration prompt. Each round produces a single output, which becomes the input to the next round, without detector feedback or candidate selection. We use the same prompt across all evaluated models and watermarking methods:

Regeneration Prompt

Re-synthesize this image as a newly rendered image. Preserve all subjects, identities, objects, positions, geometry, composition, colors, lighting, style, text, and meaningful visual details. Do not copy incidental pixel-level texture, grain, compression artifacts, sensor noise, or imperceptible high-frequency patterns from the input.

We evaluate four regeneration rounds to examine how additional rounds affect watermark removal and image fidelity.

Table 1: Watermark removal rates (%, ↑) by category. Our four-round GPT-Image-2 attack achieves the highest average removal across the 13 post-hoc and in-generation methods, while content-based watermarks remain resistant to the evaluated attacks.

	Baseline attacks									Ours (round 4)
Watermark family	Ctrl Regen	Un Marker	RINSE 4×	RINSE 2×	FLUX I2I	Diff. regen.	VAE Bmshj	VAE Cheng	DIP	GPT-Image-2
Post-hoc (7)	**97.1**	62.0	80.8	71.2	47.0	39.7	37.6	32.9	16.2	92.9
In-generation (6)	76.5	**95.5**	58.8	53.7	19.7	20.2	13.2	8.0	0.2	84.5
Non-content-based avg. (13)	87.6	77.4	70.6	63.1	34.4	30.7	26.3	21.4	8.8	**89.0**
Content-based (2)	1.5	1.5	11.1	1.5	0.5	0.5	0.5	0.0	0.0	1.5

3.2 Experimental Setup

Watermarking methods and attack baselines. We evaluate our regeneration attack against 16 watermarking configurations: eight post-hoc methods, six in-generation methods, and two content-based variants, IConMark and IConMark+TrustMark. We compare against six removal attack baselines, evaluating multiple variants and attack strengths to capture their removal–fidelity trade-offs. We follow the settings provided in the released implementations, except for the detection thresholds and model substitutions described below. Appendix A describes the baseline attacks, the parameters varied, and the number of configurations evaluated.

Metrics. We measure attack success using removal rate at detection thresholds set to a target FPR of 1%. Removal rate measures the fraction of initially detected watermarked images for which the attack evades detection. We assess fidelity relative to the watermarked input using DreamSim (Fu et al., 2023) for perceptual similarity and cosine similarities between DINOv2 (Oquab et al., 2023) and CLIP (Radford et al., 2021) image embeddings for complementary measures of visual and semantic similarity.

Datasets and detection thresholds. We evaluate each watermarking configuration on 100 images. For post-hoc methods, we use images sampled from the `laion2b-squareish-1024px` subset released by Open Diffusion AI (Open Diffusion AI), derived from LAION-5B (Schuhmann et al., 2022), and 50 AI-generated images produced by Midjourney (Midjourney). For in-generation methods, we use 100 prompts from the Stable Diffusion Prompts dataset (Gustavosta). We use detection thresholds corresponding to a target 1% FPR from the original papers or released code when available; otherwise, we calibrate thresholds using 1,000 unwatermarked images. Threshold sources, calibration details, and any exceptions are provided in Appendix B.

Models. We retain the model backbones used in the released implementations of the watermarking methods and attack baselines, except that we instantiate IConMark and IConMark+TrustMark with FLUX.2-flex (Black Forest Labs, 2025) as their image generator. Appendix A documents these adaptations. Our main evaluation uses GPT-Image-2 (OpenAI, 2026c). To examine whether comparable removal and fidelity can be achieved with lower-cost TI2I models, we evaluate two additional generators in Appendix A.1.

3.3 Attack Effectiveness and Image Fidelity

Figure 1 compares the removal-fidelity trade-offs averaged over 13 post-hoc and in-generation watermarking methods. Despite its simplicity, regeneration with GPT-Image-2 outperforms most baselines and achieves a trade-off comparable to leading attacks such as CtrlRegen. The family-level results in Table 1 provide a complementary view: CtrlRegen achieves higher removal on post-hoc watermarks but lower removal on in-generation watermarks than our attack, while UnMarker shows the opposite pattern. With GPT-Image-2, our attack achieves the highest average removal across the 13 methods among the configurations reported in Table 1, demonstrating effectiveness across both families without knowledge of the watermarking method. The appendix provides detailed per-watermark results in Tables 4–9 and additional qualitative examples in Figures 5–7. Table 1 also

reveals a distinction between watermark families. While post-hoc and in-generation watermarks are broadly vulnerable to the evaluated attacks, **the content-based watermarks remain resistant**. These results motivate content-based watermarking. We next examine IConMark, an early approach in this direction, and show how its fixed concept database enables concept extraction.

3.4 Practical Implications

Our experiments with GPT-Image-2 demonstrate that effective watermark removal is possible through API access and a fixed prompt, without local model deployment, training, or per-image optimization. Using OpenAI's published API rates (OpenAI, 2026a), we estimate the cost of four regeneration rounds at approximately $0.579 per watermarked image under our experimental settings. One practical limitation is model refusal: image-generation safety systems may reject policy-compliant requests (OpenAI, 2025) and block inputs or outputs under content policies (OpenAI, 2026e). Such refusals affected only a small fraction of our evaluated samples and were counted as attack failures. Unlike attacks that use trained control modules (Liu et al., 2025) or per-image optimization (Kassis & Hengartner, 2025; Liang et al., 2025), our approach requires only API access and a fixed prompt. Improvements in fidelity, speed, and cost could make regeneration attacks more effective and enable automated agents to remove watermarks from web images at scale. This motivates safeguards against watermark-removal misuse that preserve legitimate image editing.

4 Attacking IConMark through Concept Extraction

Our regeneration experiments show that content-based watermarks resist attacks that remove many invisible watermarks. We examine IConMark (Sadasivan et al., 2025), an early approach using human-interpretable image concepts. It maintains a fixed database $\mathcal{D}$ of 100 concepts, intended to remain private. An LLM selects nine scene-compatible concepts per prompt and adds them to the generation prompt. A VLM detects the watermark by checking all concepts in $\mathcal{D}$ and comparing the detected count against a threshold. Concept insertion reduces alignment with the original prompt. Comparing GPT-Image-2 with and without IConMark on 292 matched prompts under identical generation settings, mean CLIP similarity drops from 0.366 to 0.313 with ViT-g/14 (Cherti et al., 2023) and from 0.344 to 0.306 with ViT-B/32 (Radford et al., 2021), relative reductions of 14.4% and 11.3%. Similarity decreases on 87% and 90% of prompts, respectively. Beyond this alignment cost, database reuse enables concept extraction. We show how an adversary can recover concepts from prompts and generated images alone, then use them for watermark removal and spoofing (Figure 2).

4.1 Concept Extraction Attack

The adversary aims to recover concepts from IConMark's database $\mathcal{D}$, treated as private in deployment, using only prompt queries and the resulting images. The attack requires neither the database nor per-image concept selections or access to the watermark detector. Our central observation is that watermark concepts may appear repeatedly across independent generations, even when they are neither requested nor expected from the prompt.

1. Constructing candidate concepts. For each prompt $p \in \mathcal{P}$, we obtain m independent images from the watermarked generator. An attacker-controlled VLM lists the objects in each image without seeing the prompt. We use an LLM to identify objects explicitly named in the prompt and check each proposed exclusion in a second pass. We exclude an object only if both passes agree that it is explicitly named in the prompt. To consolidate naming variations, we propose groups using shared head nouns and embedding neighborhoods, then use an LLM to identify equivalent names. We merge two groups only if no pair of names across them has been identified by the LLM as referring to different objects. The resulting groups form our candidate set $\mathcal{C}$.

2. Measuring unexpectedness and persistence. For each prompt p, we ask an LLM several times to list the objects it would expect to see in the image, without showing it any generated images. Objects listed in at least k responses form the expected-object set $\mathcal{E}_p$. For candidate c, we measure its maximum similarity to this list:

$$a(c,p) = \max_{o \in \mathcal{E}_p} \cos(\mathbf{h}_c, f(o)), \qquad u(c,p) = \max\left\{\sigma\left(\frac{\mu_p - a(c,p)}{2s_p}\right), 0.05\right\}, \tag{2}$$

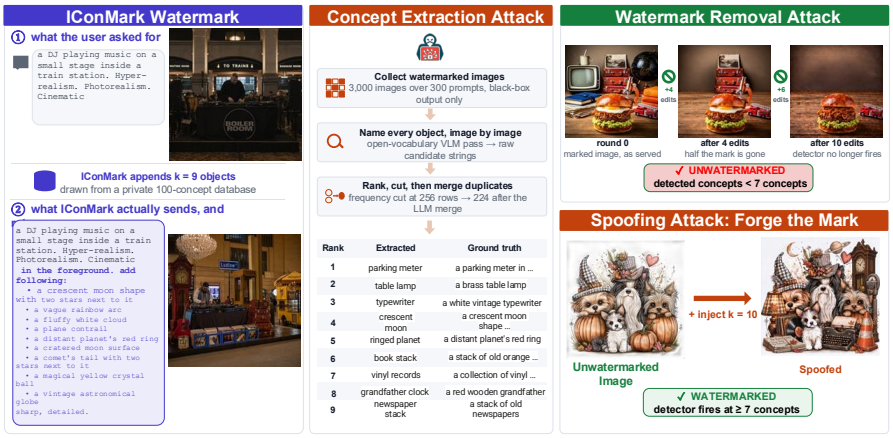

Figure 2: Overview of IConMark and our attacks.

where f is a text encoder, $\mathbf{h}_c$ is the centroid of the candidate's name embeddings, and σ is the logistic function. The mean μ_p and standard deviation s_p are estimated by scoring randomly sampled candidate centroids against the same expected-object list. This normalization accounts for prompt-dependent similarity distributions, including differences in list length.

An unexpected object may appear in an image by chance. To measure whether it appears consistently across generations, we define persistence $r(c, p)$ as the fraction of the m images generated from prompt p whose extracted object lists contain candidate c. We combine names that may refer to the same object when they tend to appear in different images generated from the same prompt. Each pooled group contributes once, using its combined image coverage and the mention-weighted mean unexpectedness of its members. The evidence for a candidate is

$$e(c, p) = r(c, p)\, u(c, p), \tag{3}$$

favoring concepts that are both persistent and unexpected.

3. Ranking and selecting candidates. We retain candidates observed under at least two distinct prompts and rank them by the mean of their two highest prompt-level evidence scores:

$$S(c) = \frac{e_{(1)}(c) + e_{(2)}(c)}{2}, \tag{4}$$

where $e_{(1)}(c) \geq e_{(2)}(c)$. To set the selection threshold, we apply the same extraction and scoring procedure to three unwatermarked generators. We use the largest of their within-generator 95th-percentile candidate scores as the threshold τ and retain candidates satisfying $S(c) > \tau$. Finally, we consolidate duplicate names only when two independently shuffled LLM passes agree, requiring pairwise agreement within each merged group. The resulting ranked concept list supports the removal and spoofing attacks described below.

4.2 Experimental Evaluation

Watermark system. We use IConMark's original database of 100 concepts and follow its setting of selecting nine concepts per input prompt. We use GPT-5.6-Terra (OpenAI, 2026b) as the concept selector, IDEFICS3-8B-Llama3 (Laurençon et al., 2024) as the watermark detector, and GPT-Image-2 with medium quality as the image generator. The attacker has no access to the database; comparisons against it are performed only during evaluation.

Attack models and settings. We select 300 cleaned, deduplicated Midjourney user prompts using text embeddings for topic diversity and generate ten 1024×1024 images per prompt (3,000). Claude Opus 5 (Anthropic, 2026) performs object extraction and text-based judgments; MiniLM-L6-v2 scores unexpectedness. We propose candidate groups using 12 nearest neighbors in BGE-large-en-v1.5 embedding space, with pooling requiring MiniLM cosine similarity ≥ 0.75. Expected

objects must appear in at least four of six responses per prompt ($k = 4$). We calibrate the candidate-score threshold on unwatermarked images from FLUX.2-flex, MAI-Image-2.5, and Qwen-Image-2512 (Wu et al., 2025) using 100 shared prompts, taking the largest 95th percentile across generators. We report database coverage, the fraction of concepts recovered, using an LLM judge with agreement across two shuffled passes and one-to-one matching to avoid double-counting concepts.

Concept extraction results. Using 3,000 watermarked images from 300 prompts, our attack recovers 79 of the 100 database concepts. The calibrated threshold retains 256 of the 3,685 scored candidates. Merging duplicate descriptions reduces this list to 224 candidates without changing coverage, yielding 79% recall and 35.3% precision under one-to-one matching. These results show that reuse of a fixed concept database allows an adversary to recover most of its concepts using only input prompts and generated images, without access to the database or watermark detector. The list contains redundant descriptions and candidates that do not match database entries. In the following subsection, we evaluate whether this list is sufficient to support watermark removal and spoofing.

4.3 Removal and Spoofing with Extracted Concepts

Watermark removal. We test whether the 224 candidates extracted from a GPT-Image-2-based deployment transfer to 100 IConMark-watermarked images from the FLUX.2 deployment. Claude Opus 5 identifies candidates present in each initial image; we sequentially remove the ten highest-ranked using GPT-Image-2.5-Sunburst (OpenAI, 2026d) at high quality, instructing it to preserve other content. The target order is fixed before editing, with no detector feedback. We reuse the images from Section 3 and evaluate both removal and spoofing with its IDEFICS3-8B-Llama3 detector and threshold of at least seven concepts, ensuring consistent comparison with regeneration. Removal is measured over the 99 initially detected images; the refused chain counts as a failure and is excluded from fidelity averages. Removal reaches 51.5% after four edits and 93.9% (93/99) after ten (Figure 9). After ten edits, mean CLIP similarity, DreamSim distance, and DINOv2 similarity to the watermarked input are 0.812, 0.342, and 0.566, respectively, showing that targeted removal comes at a cost to image fidelity.

Watermark spoofing. Spoofing makes images created outside a watermarking system pass its detector (Saberi et al., 2024; Müller et al., 2025). We use the candidates to edit 27 unwatermarked Midjourney images and generate images from 27 fixed LLM-written scene prompts. For each input, we request zero to ten highest-ranked concepts absent from its initial image, using GPT-Image-2.5-Sunburst at high quality without detector feedback. Editing inserts concepts cumulatively; generation appends their descriptions to the prompt and uses a separate sample at each budget (Figure 4). With the same detector and threshold as above, ten concepts yield spoofing rates of 59.3% (16/27) for editing and 70.4% (19/27) for generation, counting refusals as failures. Neither setting triggers detection with zero concepts. After ten insertions, edited images have mean CLIP similarity, DreamSim distance, and DINOv2 similarity of 0.851, 0.207, and 0.546 relative to the originals.

5 Image-Specific Content-Based Watermarking

Our regeneration experiments suggest that meaningful image content can provide a basis for robust watermarking. However, our analysis of IConMark shows that reusing a fixed concept database enables concept extraction and subsequent removal and spoofing. We therefore propose K-SemMark-R, an in-generation watermark that uses randomly selected, prompt-compatible scene details as carriers. These carriers are constructed separately for each prompt, and their realized values are stored with the generated image's embedding in a private registry. Detection retrieves candidate records and checks their stored carrier values against the query image. Figure 3 illustrates the enrollment and detection procedure.

5.1 Methodology

Carrier planning. Given a prompt p, an LLM proposes carrier slots, each describing a scene attribute that the prompt leaves unspecified. Each slot s_j has an alphabet $\mathcal{A}_j$ of mutually exclusive values and a rendering instruction for each value. A lantern's material could be brass or copper. We filter candidates for prompt compatibility, alphabet size, and dependence between slots, then

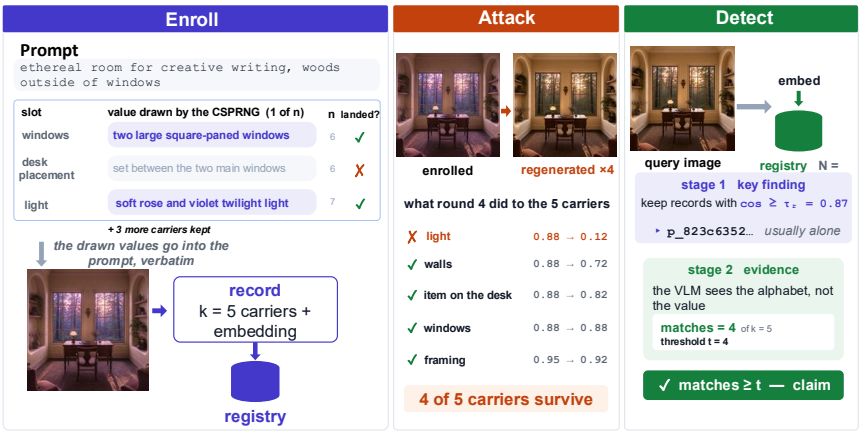

Figure 3: Overview of K-SemMark-R.

privately select a subset. *We attempt watermarking only when the prompt admits enough compatible carriers; otherwise, we abstain rather than introduce unrelated content to embed a watermark.*

Generation and enrollment. For each selected slot, we sample a value $v_j \in \mathcal{A}_j$ using a cryptographically secure random number generator and append its rendering instruction to the prompt. After image generation, a VLM scores every alternative in each slot without being told which value was requested. We retain carriers whose sampled values receive confidence at least γ_e. Enrollment requires at least three retained carriers; we allow up to two generation attempts using the same carrier configuration. For a successfully enrolled image I, the private registry stores its embedding $f(I)$ and the retained slots, alphabets, and sampled values.

Table 2: Removal rate (%) for the 13 non-content-based methods (equal-weight mean) and K-SemMark-R.

Attack	Other methods (13)	Ours
CtrlRegen	87.6	35.3
UnMarker	77.4	25.0
GPT-Image-2	89.0	14.7

Detection. Given a query image J, we retrieve every registry record whose stored embedding has cosine similarity at least τ_r to $f(J)$. Embedding similarity identifies candidates but does not by itself establish watermark presence. For each candidate record R, the VLM scores the full alphabet of each stored slot, without being told the sampled value. For a record with k carriers, we count

$$S(J,R) = \sum_{j=1}^{k} \mathbf{1}[q_j(v_j; J) \geq \gamma_d], \tag{5}$$

where $q_j(v_j; J)$ is the VLM confidence that the stored value v_j is present in J. We choose a threshold $t(k)$ separately for each record as the smallest integer meeting a target per-record FPR of $\alpha = 0.01$ under a null model calibrated on unwatermarked images. The query is detected as watermarked if at least one retrieved record satisfies $S(J,R) \geq t(k)$. The query is detected as watermarked if at least one retrieved record satisfies $S(J,R) \geq t(k)$. The per-record calibration target does not by itself establish an image-level FPR bound for the full registry.

5.2 Experimental Evaluation

Experimental Setup. We use GPT-Image-2 for generation, Claude Opus 5 for enrollment verification and detection, and CLIP ViT-g/14 for retrieval. On the 100 prompts from Section 3, we evaluate CtrlRegen and UnMarker, the two strongest baselines by average removal in Table 1, and four GPT-Image-2 regeneration rounds. Attack results evaluate carrier verification; retrieval is evaluated separately. Appendix C.1 provides detailed settings and evaluation procedures.

Enrollment and clean detection. K-SemMark-R enrolls 75 of 100 prompts, retaining an average of 4.88 carriers per image. Of these, 68 are detected before attack: 90.7% of enrolled images and

68.0% of all prompts. The 25 unsuccessful enrollments comprise 18 carrier-capacity or verification failures, six generator refusals, and one verifier-response parsing failure. In a separate retrieval evaluation, $\tau_r = 0.87$ retains the correct record for 95.9% of attacked queries. The complete detector produces one false acceptance among 4,499,972 unenrolled Midjourney images (Naseh et al., 2024) with the evaluated registry.

Robustness to baseline attacks. CtrlRegen and UnMarker achieve removal rates of 35.3% (24/68) and 25.0% (17/68), compared with 87.6% and 77.4% averaged over the 13 non-content-based methods under the same configurations (Table 2). Mean DreamSim distances are 0.084 and 0.088 for our images, versus 0.083 and 0.074 for those methods. The lower removal rates apply to these configurations; different image pools, generators, and resolutions limit broader comparisons. Four regeneration rounds of our regeneration attack remove 14.7% (10/68) of initially detected watermarks, with mean DreamSim distance 0.056, DINOv2 similarity 0.937, and CLIP similarity 0.963. Re-verifying unchanged images reverses 7.4% (5/68) of positive decisions, close to the 8.8% (6/68) removal after one round, indicating that verifier variability contributes to observed detection changes. On the separate 200-image development set, carrier detection falls from 89.5% before attack to 82.0% with the fixed prompt and 79.5% with the less constrained prompt after four rounds, counting incomplete sequences as failed attacks. This suggests sensitivity to regeneration instructions.

Carrier recurrence and concept extraction. Unlike IConMark's shared inventory, K-SemMark-R uses prompt-conditioned alphabets and independently sampled carrier values. Ten generation attempts per prompt, with fixed carrier plans across ten prompts, yield 79 enrolled images across nine prompts. Applying the extraction attack's matching rules to enrolled carrier descriptions gives mean persistence 0.190, unchanged after pooling naming variants, computed over enrolled images per prompt. Across 357 records, 67 of 1,748 carrier instances (3.8%) share normalized values with another record. All 366 carrier values in the separate 75-image evaluation set are distinct after string normalization. These results show limited recurrence within and across prompts, supporting resistance to reusable-dictionary extraction, but do not establish failure of the full extraction pipeline or resistance to adaptive attacks on individual images.

Text-image alignment. Using the CLIP checkpoints from Section 4, mean similarity to the original prompt is 0.366 (ViT-g/14) and 0.324 (ViT-B/32) on the 75 enrolled images, and 0.352 and 0.325 on the 357-image development set. These scores cover enrolled images only; different prompt sets prevent attributing the higher scores relative to IConMark solely to the watermarking method.

Discussion. K-SemMark-R demonstrates an approach to content-based watermarking that uses image-specific carriers without a shared concept inventory. The measured recurrence and carrier-verification results support further study of this design, while incomplete enrollment and verifier variability remain practical limitations. The method could be combined with a post-hoc watermark, following prior work on hybrid watermarking (Sadasivan et al., 2025). Such a combination could provide complementary detection signals, but its robustness and FPR would require joint evaluation.

6 Conclusion

We show that GPT-Image-2 can remove many evaluated watermarks using only API access and a fixed prompt, with a removal-fidelity trade-off comparable to specialized attacks. The evaluated content-based watermarks resist regeneration, but IConMark's fixed concept inventory enables extraction, targeted removal, and spoofing. We therefore introduce K-SemMark-R, which uses prompt-compatible, image-specific carriers stored in a private registry. Its limited carrier recurrence and resistance to the tested attacks at the carrier-verification stage support this design, while prompt coverage, verifier variability, and end-to-end robustness require further study. These findings motivate continued research into content-based watermarking that preserves detectability under regeneration without relying on a shared concept inventory.

AI USE STATEMENT

In this work, we used generative AI tools to assist with research methodology and experiment design, qualitative failure analysis, and interpretation of experimental results. Generative AI was also used to assist with drafting and editing the manuscript. We did not use generative AI to generate synthetic datasets, formulate or prove mathematical claims, or translate the paper. All AI-assisted outputs were reviewed and verified by the authors against the underlying experiments and data. We take responsibility for the final content of this work.

REFERENCES

Bang An, Mucong Ding, Tahseen Rabbani, Aakriti Agrawal, Yuancheng Xu, Chenghao Deng, Sicheng Zhu, Abdirisak Mohamed, Yuxin Wen, Tom Goldstein, et al. Waves: Benchmarking the robustness of image watermarks. *arXiv preprint arXiv:2401.08573*, 2024.

Anthropic. Introducing Claude Opus 5, 2026. URL https://www.anthropic.com/news/claude-opus-5.

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, 2025.

Black Forest Labs. FLUX.2: Frontier visual intelligence, 2025. URL https://bfl.ai/blog/flux-2.

Tu Bui, Shruti Agarwal, and John Collomosse. Trustmark: Universal watermarking for arbitrary resolution images. *arXiv preprint arXiv:2311.18297*, 2023.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2829. IEEE, 2023.

Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. Watermarking images in self-supervised latent spaces. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3054–3058. IEEE, 2022.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22409–22420. IEEE, 2023.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*, 2023.

Sven Gowal, Rudy Bunel, Florian Stimberg, David Stutz, Guillermo Ortiz-Jimenez, Christina Kouridi, Mel Vecerik, Jamie Hayes, Sylvestre-Alvise Rebuffi, Paul Bernard, et al. Synthid-image: Image watermarking at internet scale. *arXiv preprint arXiv:2510.09263*, 2025.

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

Gustavosta. Stable-Diffusion-Prompts. Hugging Face dataset. URL https://huggingface.co/datasets/Gustavosta/Stable-Diffusion-Prompts.

Huayang Huang, Yu Wu, and Qian Wang. Robin: Robust and invisible watermarks for diffusion models with adversarial optimization. *Advances in Neural Information Processing Systems*, 37: 3937–3963, 2024.

Andre Kassis and Urs Hengartner. Unmarker: A universal attack on defensive image watermarking. In *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 2602–2620. IEEE, 2025.

Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. Building and better understanding vision-language models: insights and future directions. *arXiv preprint arXiv:2408.12637*, 2024.

Hengyue Liang, Taihui Li, and Ju Sun. A baseline method for removing invisible image watermarks using deep image prior. *Transactions on machine learning research*, 2025:4248, 2025.

Yepeng Liu, Yiren Song, Hai Ci, Yu Zhang, Haofan Wang, Mike Zheng Shou, and Yuheng Bu. Image watermarks are removable using controllable regeneration from clean noise. In *International Conference on Learning Representations*, volume 2025, pp. 87310–87327, 2025.

Shilin Lu, Zihan Zhou, Jiayou Lu, Yuanzhi Zhu, and Adams Kong. Robust watermarking using generative priors against image editing: From benchmarking to advances. In *International Conference on Learning Representations*, volume 2025, pp. 83902–83936, 2025.

Microsoft AI. Introducing MAI-Image-2.5-Pro and MAI-Voice-2-Flash, 2026. URL https://microsoft.ai/news/introducing-mai-image-2-5-pro-and-mai-voice-2-flash/.

Midjourney. Midjourney. URL https://www.midjourney.com/.

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

Ali Naseh, Katherine Thai, Mohit Iyyer, and Amir Houmansadr. Iteratively prompting multimodal llms to reproduce natural and ai-generated images. *arXiv preprint arXiv:2404.13784*, 2024.

KA 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. IEEE, 2008.

Open Diffusion AI. laion2b-squareish-1024px. Hugging Face dataset. URL https://huggingface.co/datasets/opendiffusionai/laion2b-squareish-1024px.

OpenAI. Addendum to GPT-4o system card: Native image generation, March 2025. URL https://cdn.openai.com/11998be9-5319-4302-bfbf-1167e093f1fb/Native_Image_Generation_System_Card.pdf.

OpenAI. OpenAI API Pricing, 2026a. URL https://developers.openai.com/api/docs/pricing.

OpenAI. GPT-5.6 Terra, 2026b. URL https://developers.openai.com/api/docs/models/gpt-5.6-terra.

OpenAI. GPT-Image-2, 2026c. URL https://developers.openai.com/api/docs/models/gpt-image-2.

OpenAI. GPT-Image-2.5 Sunburst, 2026d. URL https://developers.openai.com/api/docs/models/gpt-image-2.5-sunburst.

OpenAI. ChatGPT Images 2.0 system card, April 2026e. URL https://deploymentsafety.openai.com/chatgpt-images-2-0.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Xavier Pleimling, Sifat M_Abdullah, Gunjan Balde, Peng Gao, Mainack Mondal, Murtuza Jadliwala, and Bimal Viswanath. Off-the-shelf image-to-image models are all you need to defeat image protection schemes. 2026.

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

Mehrdad Saberi, Vinu Sankar Sadasivan, Keivan Rezaei, Aounon Kumar, Atoosa Chegini, Wenxiao Wang, and Soheil Feizi. Robustness of ai-image detectors: Fundamental limits and practical attacks. In *International Conference on Learning Representations*, volume 2024, pp. 27500–27526, 2024.

Vinu Sankar Sadasivan, Mehrdad Saberi, and Soheil Feizi. Iconmark: Robust interpretable concept-based watermark for ai images. *arXiv preprint arXiv:2507.13407*, 2025.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

Fahad Shamshad, Nils Lukas, and Karthik Nandakumar. Raven: Erasing invisible watermarks via novel view synthesis. *arXiv preprint arXiv:2601.08832*, 2026.

Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2114–2123. IEEE, 2020.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *arXiv preprint arXiv:2305.20030*, 2023.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025. URL `https://arxiv.org/abs/2508.02324`.

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.

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

Lijun Zhang, Xiao Liu, Antoni V Martin, Cindy X Bearfield, Yuriy Brun, and Hui Guan. Attack-resilient image watermarking using stable diffusion. *Advances in Neural Information Processing Systems*, 37:38480–38507, 2024.

Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative ai. *Advances in neural information processing systems*, 37:8643–8672, 2024.

A Regeneration Attack

Attack baselines. We evaluate seven baseline attacks: VAE compression and diffusion regeneration (Regen-VAE and Regen-Diff; Zhao et al., 2024), iterative regeneration (Rinse; An et al., 2024), deep-image-prior reconstruction (DIP; Liang et al., 2025), FLUX-based image-to-image denoising (FLUX-img2img; Pleimling et al., 2026), optimization-based removal (UnMarker; Kassis & Hengartner, 2025), and controllable regeneration (CtrlRegen; Liu et al., 2025). We use the official implementations at fixed repository commits and follow the released settings except where noted.

To characterize removal-fidelity trade-offs, we vary compression quality for Regen-VAE (12 configurations across two learned codecs), the diffusion noise step for Regen-Diff (5 configurations), the number of passes and noise step for Rinse (8 configurations), the stopping iteration for DIP (11 configurations), and regeneration strength for CtrlRegen (13 configurations across CtrlRegen and CtrlRegen+). For FLUX-img2img, we vary denoising strength and compare four instruction prompts at a fixed strength, yielding 10 configurations in total. We evaluate UnMarker using one fixed configuration from its released implementation. Together, these settings yield 60 baseline attack configurations.

A.1 Effect of the Regeneration Model

We examine whether the same regeneration procedure achieves comparable removal and fidelity with FLUX.2-flex and MAI-Image-2.5-Pro (Microsoft AI, 2026). We use the same watermarked inputs, fixed prompt, four-round procedure, and evaluation protocol as in Section 3. Figure 8 compares the three models, with per-watermark removal and fidelity reported in Tables 4 and 5. After four rounds, FLUX.2-flex and MAI-Image-2.5-Pro achieve average removal rates of 84.6% and 52.0%, respectively, across the 13 non-content-based watermarking methods, compared with 89.0% for GPT-Image-2. Although FLUX.2-flex approaches GPT-Image-2 in removal, it produces larger changes to the input under all three fidelity metrics. MAI-Image-2.5-Pro retains slightly higher average fidelity than GPT-Image-2 but achieves substantially lower removal. For all three models, additional rounds increase average removal while reducing fidelity (Tables 6 and 7–9). These results show that the removal–fidelity trade-off depends on the regeneration model: API access and a fixed prompt can enable effective removal, but the same procedure does not perform equally well across models.

B Watermark Detection and Threshold Selection

This section describes the detection rules used to evaluate watermark removal. We use a target FPR of 1% and obtain thresholds from published statistical tests, released implementations, or calibration on unwatermarked images. The resulting rules are fixed across all attack methods, strengths, and regeneration rounds. Table 3 summarizes the thresholds used in our experiments and their observed false positives. IConMark+TrustMark uses the combined decision rule described below.

B.1 Detection and Calibration Protocol

When a watermark's published test or released implementation accepts an FPR parameter, we evaluate that rule at $\alpha = 0.01$. We also derive thresholds from the binomial test described in W-Bench for TrustMark and StegaStamp (Lu et al., 2025). For the empirically calibrated methods, we score 1,000 unwatermarked images per method and select a threshold that admits at most ten false positives. The negative pools consist of real images for post-hoc methods, unwatermarked Stable Diffusion 2.1 images for in-generation methods, and unwatermarked FLUX.2-flex images for the IConMark configurations. The pools are matched to the evaluation resolution.

Detection requires a score above the threshold for similarity and match-count statistics, and below the threshold for distance statistics. We use the inclusive comparisons listed in Table 3. During calibration, boundary ties are excluded when including them would exceed the false-positive budget. For discrete scores, we report the equivalent minimum number of matching bits or concepts, rather than the floating-point boundary stored by the implementation.

The target FPR and the observed FPR need not coincide. Discrete statistics may not admit a threshold that gives exactly 1%, and published tests rely on their stated null models or probability bounds. We therefore report the false-positive counts measured on the 1,000-image negative pools, including for thresholds derived from published rules. These counts describe the finite evaluation pools; observing no false positives does not establish a population FPR of zero.

B.2 Thresholds Derived from Published Tests and Code

Stable Signature. Stable Signature (Fernandez et al., 2023) detects a watermark by comparing the decoded message with a reference signature. Under its null model, the number of matching bits M

follows $\mathrm{Binomial}(K, 1/2)$. For an inclusive detection rule, we select

$$t_K = \min\left\{ t \in \{0, \ldots, K+1\} : \sum_{j=t}^{K} \binom{K}{j} 2^{-K} \le \alpha \right\}. \tag{6}$$

For the 48-bit signature used in our experiments, $t_{48} = 33$, giving the rule $M \ge 33$, or bit accuracy at least $33/48 = 0.6875$. The binomial FPR is approximately 0.664%. This is a single-signature detection test, without a search over multiple users' signatures.

VINE, TrustMark, and StegaStamp. We use the 100-bit matching statistic for VINE-R (Lu et al., 2025), TrustMark (Bui et al., 2023), and StegaStamp (Tancik et al., 2020). Following the statistical test in W-Bench (Lu et al., 2025, Appendix C), we apply Equation 6 with $K = 100$. The resulting rule is $M \ge 63$, or bit accuracy at least 0.63, with a binomial FPR of approximately 0.602%. This threshold is derived from the published test; it is not a numerical threshold reported in a results table. For TrustMark and StegaStamp, matches are measured on the raw decoded bits before BCH error correction. The observed false-positive counts are seven, six, and five out of 1,000 for VINE-R, TrustMark, and StegaStamp, respectively.

SSL Watermarking. We use the zero-bit detector of SSL Watermarking (Fernandez et al., 2022). Let c be the absolute cosine similarity between the image feature and the secret carrier, and let d be the feature dimension. The published test assigns

$$p_{\mathrm{SSL}} = I_{1-c^2}\left(\frac{d-1}{2}, \frac{1}{2}\right), \tag{7}$$

where $I_x(a, b)$ is the regularized incomplete beta function. Our detector stores $-\log_{10} p_{\mathrm{SSL}}$ and declares a watermark when this score is at least 2, equivalently $p_{\mathrm{SSL}} \le 0.01$. For $d = 2048$, this corresponds to $c \gtrsim 0.056893$. The threshold is applied at detection time without changing the embedding parameters of the watermarked images.

Gaussian Shading. We use the threshold returned by the official Gaussian Shading implementation (Yang et al., 2024) with its FPR parameter set to 0.01. With channel and spatial replication factors of 1 and 8, respectively, the decoded watermark has 256 bits. The implementation returns a bit-accuracy threshold of $147/256 = 0.57421875$ and uses an inclusive comparison. We retain this rule in all experiments. The code computes a strict binomial tail when selecting the threshold but uses an inclusive comparison when applying it; consequently, the binomial FPR of the implemented rule is approximately 1.028%. On our negative pool, the observed FPR is $6/1000 = 0.6\%$.

PRC. We use PRC's statistical detector (Gunn et al., 2025), with the FPR parameter set to $F = 0.01$. To express its image-dependent comparison as a fixed threshold, let u_j be the product of the adjusted bit posteriors for parity check j, as defined by the detector, and write

$$a_j = \ln \frac{1+u_j}{2}, \qquad b_j = \ln \frac{1-u_j}{2}, \qquad C = \frac{1}{2}\sum_j (a_j - b_j)^2. \tag{8}$$

For $C > 0$, the published test is equivalent to

$$s_{\mathrm{PRC}} = \frac{\sum_j a_j - \frac{1}{2}\sum_j (a_j + b_j)}{\sqrt{C}} \ge \sqrt{\ln(1/F)}. \tag{9}$$

Thus, the threshold on our normalized score is $\sqrt{\ln 100} = 2.145966026289347$. This is an equivalent form of the published detector, rather than an empirically fitted threshold. It produces two false positives on the 1,000-image negative pool.

ZoDiac. ZoDiac (Zhang et al., 2024) computes a p-value from a noncentral chi-squared test on the inverted image latent. Our implementation uses $1 - p$ as its score and declares a watermark when $1 - p \ge 0.99$, corresponding to a nominal 1% significance level. The paper uses a strict comparison and reports an empirical FPR of 0.4% at this setting. Our negative pool also gives $4/1000 = 0.4\%$.

B.3 Empirically Calibrated Thresholds

DWT–DCT–SVD and RivaGAN. We evaluate the blind 32-bit `dwtDctSvd` implementation in `invisible-watermark`[1] and the 32-bit RivaGAN decoder (Zhang et al., 2019). For both methods, the score is the fraction of decoded bits matching the reference payload. Calibration gives a minimum of 23 matching bits, or bit accuracy at least $23/32 = 0.71875$. The observed false-positive counts are eight for DWT–DCT–SVD and five for RivaGAN. Both rules are stored as a floating-point boundary just above $22/32$; reporting the boundary as $22/32$ would incorrectly include another attainable score.

Tree-Ring. For Tree-Ring (Wen et al., 2023), we use the noncentral chi-squared p-value statistic provided in the released implementation. This differs from the distance statistic used in the paper's main detection rule. Smaller p-values indicate a closer match to the watermark. Calibration on unwatermarked Stable Diffusion 2.1 images gives $\tau_{\mathrm{TR}} = 0.018857631669603622$. We detect a watermark when $p_{\mathrm{TR}} \leq \tau_{\mathrm{TR}}$, which admits ten false positives out of 1,000.

ROBIN. ROBIN (Huang et al., 2024) is evaluated using the mean complex ℓ_1 distance between the recovered latent watermark and its reference over the watermark mask. We use the statistic evaluated at diffusion step 35. Lower distances indicate watermark presence. The calibrated rule is $d_{\mathrm{R}} \leq \tau_{\mathrm{R}}$, where $\tau_{\mathrm{R}} = 48.53644943237304$; it yields ten false positives out of 1,000.

SEAL. We use SEAL's exhaustive spatial test (Arabi et al., 2025). For each recovered latent patch, the detector tests the candidate watermark vectors and counts a match when the minimum Euclidean distance satisfies the patch tolerance of 2.3. We calibrate the threshold on this match count, keeping the patch tolerance fixed. The resulting detection rule requires at least 31 matches and gives ten false positives out of 1,000. This count threshold is specific to the exhaustive detector used in our experiments.

IConMark. IConMark (Sadasivan et al., 2025) is scored by counting the concepts from its 100-concept database that the verifier detects in an image. We calibrate the FLUX.2-flex configuration used in our removal experiments on 1,000 unwatermarked FLUX.2-flex images, using the same database and verifier as in evaluation. The resulting rule requires at least seven detected concepts and produces six false positives. Calibration is specific to this generator and verifier configuration.

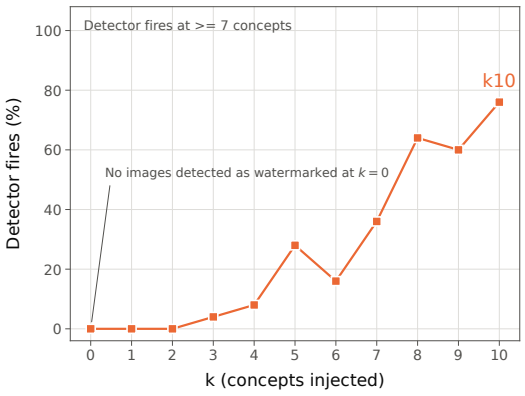

Figure 4: Spoofing IConMark by generating images with k recovered concepts added to the prompt. Results cover 25 scenes with outputs available at every concept budget. Detection requires at least seven database concepts.

B.4 IConMark+TrustMark Detection

The IConMark+TrustMark configuration combines concept detection with recovery of the expected TrustMark payload. Let C_{IC} be the detected concept count and let $T_{\mathrm{TM}} \in \{0, 1\}$ indicate successful decoding and an exact match to the expected payload. The hybrid decision used in our experiments is

$$D_{\mathrm{hybrid}}(x) = \mathbf{1}\{C_{\mathrm{IC}}(x) \geq 7 \ \vee \ T_{\mathrm{TM}}(x) = 1\}\,. \tag{10}$$

The TrustMark component of this hybrid therefore uses a payload decision, whereas standalone TrustMark uses the raw-bit threshold described above. We evaluate the combined decision on the same 1,000 unwatermarked FLUX.2-flex images, obtaining six false positives. This fixed combined rule is reported separately from the scalar thresholds; its two components were not jointly calibrated to attain exactly 1% FPR.

[1] `https://github.com/ShieldMnt/invisible-watermark`

Table 3: Detection rules used throughout the watermark-removal experiments. FP is the number of false positives on the corresponding 1,000-image negative pool; the observed FPR is therefore $\mathrm{FP}/1000$. Paper denotes a rule derived from the method's published test, Code denotes the released implementation configured for a target FPR of 1%, W-Bench denotes its published binomial test, and Calib. denotes empirical calibration. IConMark and IConMark+TM use FLUX.2-flex; TM denotes TrustMark. $\tau_{\mathrm{R}} = 48.53644943237304$ and $\tau_{\mathrm{TR}} = 0.018857631669603622$. Gaussian Shading retains the official code's inclusive comparison. The hybrid uses concept detection OR an exact decoded-payload match. Counts are measured on the reported negative pools and are not estimates from a separate validation set.

Watermark	Statistic	Detection rule	Source	FP
Post-hoc				
DWT–DCT–SVD Navas et al. (2008)	Bit accuracy	$\geq 23/32$	Calib.	8
RivaGAN Zhang et al. (2019)	Bit accuracy	$\geq 23/32$	Calib.	5
SSL Watermarking Fernandez et al. (2022)	$-\log_{10} p_{\mathrm{SSL}}$	≥ 2	Paper	7
StegaStamp Tancik et al. (2020)	Raw bit accuracy	$\geq 63/100$	W-Bench	5
TrustMark Bui et al. (2023)	Raw bit accuracy	$\geq 63/100$	W-Bench	6
VINE-R Lu et al. (2025)	Bit accuracy	$\geq 63/100$	Paper	7
ZoDiac Zhang et al. (2024)	$1 - p$	≥ 0.99	Paper	4
In-generation				
Gaussian Shading Yang et al. (2024)	Bit accuracy	$\geq 147/256$	Code	6
PRC Gunn et al. (2025)	s_{PRC}	$\geq \sqrt{\ln 100}$	Paper	2
ROBIN Huang et al. (2024)	Complex ℓ_1 distance	$\leq \tau_{\mathrm{R}}$	Calib.	10
SEAL Arabi et al. (2025)	Patch-match count	≥ 31	Calib.	10
Stable Signature Fernandez et al. (2023)	Bit accuracy	$\geq 33/48$	Paper	0
Tree-Ring Wen et al. (2023)	p_{TR}	$\leq \tau_{\mathrm{TR}}$	Calib.	10
Content-based				
IConMark Sadasivan et al. (2025)	Concept count	$C_{\mathrm{IC}} \geq 7$	Calib.	6
IConMark+TM Sadasivan et al. (2025); Bui et al. (2023)	Concept/payload	Equation 10	Combined	6

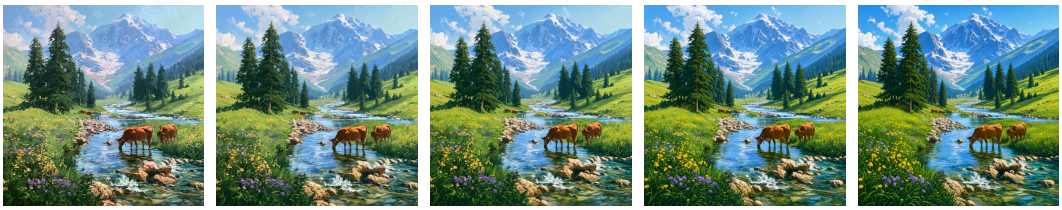

Figure 5: Four-round GPT-Image-2 regeneration of an SSL-watermarked image. The watermark is undetected from round two onward.

C Image-Specific Content-Based Watermarking

C.1 K-SemMark-R Experimental Details

Model settings and calibration. GPT-Image-2 generates images at medium quality and native 1024×1024 resolution, compared with 512×512 in the earlier watermark experiments. Claude Opus 5 uses medium reasoning effort for enrollment verification and detection. We set the CLIP ViT-g/14 retrieval threshold to $\tau_r = 0.87$ and the enrollment and detection confidence thresholds to $\gamma_e = 0.5$ and $\gamma_d = 0.6$. Record-specific thresholds are fixed at enrollment under a binomial null model with target per-record FPR $\alpha = 0.01$. The null carrier-match probability, $p_0 = 0.1375$, is estimated from 370 unwatermarked images.

Attacks and evaluation protocol. CtrlRegen and UnMarker use the configurations reported in Table 1. GPT-Image-2 regeneration uses the fixed prompt from Section 3. We report enrollment success over all input prompts, clean carrier detection over enrolled images, and removal over initially detected images. Incomplete regeneration sequences count as failed attacks. Fidelity is measured relative to the watermarked input using the metrics in Section 3.2. For attack evaluation, carrier verification uses the corresponding enrolled record. We repeat verification of unchanged images to measure detector variability. On a separate set of 200 enrolled development images, we compare the fixed regeneration prompt with a shorter version without explicit content-preservation instructions.

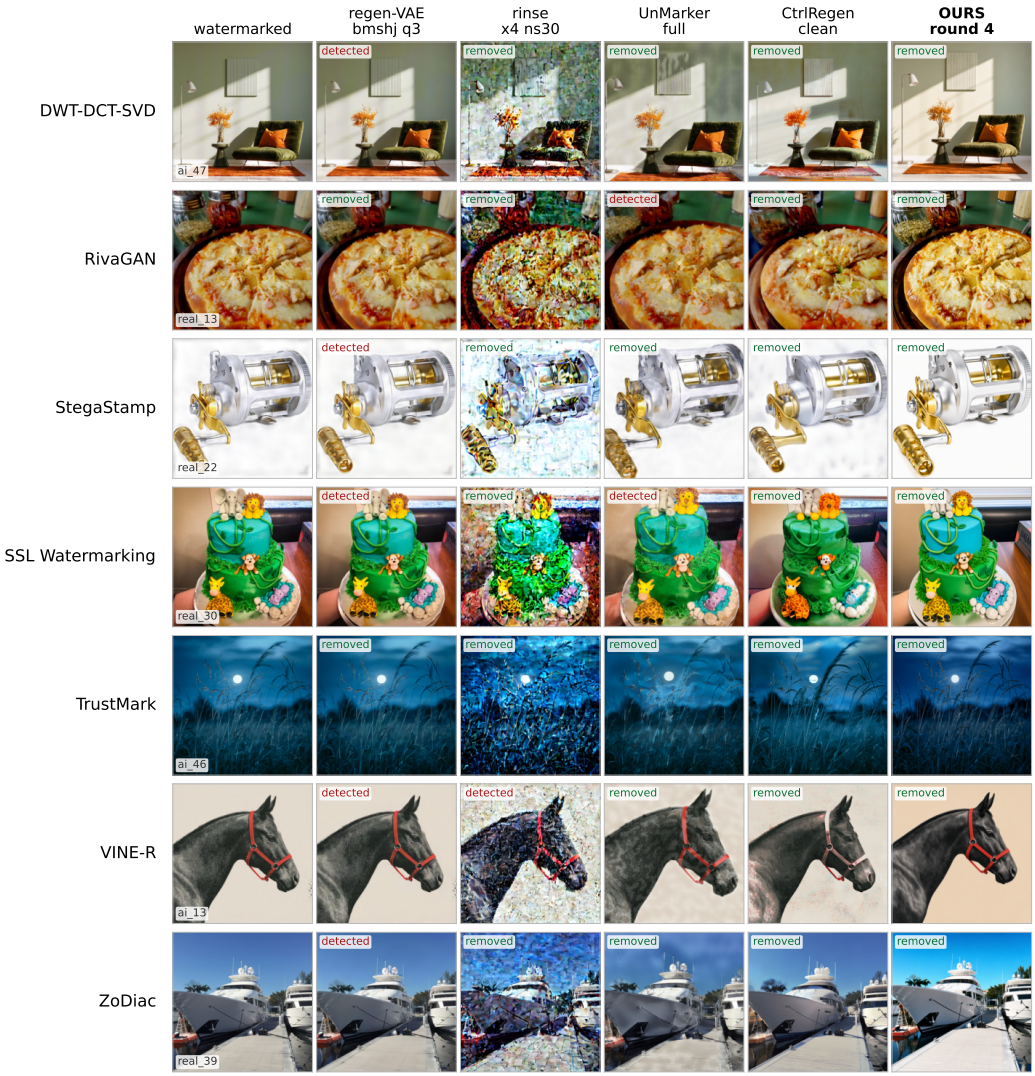

Figure 6: Qualitative comparison of watermark-removal attacks on seven post-hoc watermarking methods. Each row shows a different watermarked input, outputs from four baseline attacks, and our GPT-Image-2 regeneration output after four rounds. Green "removed" labels indicate successful evasion of the watermark detector; red "detected" labels indicate that the watermark remains detectable under the evaluation thresholds.

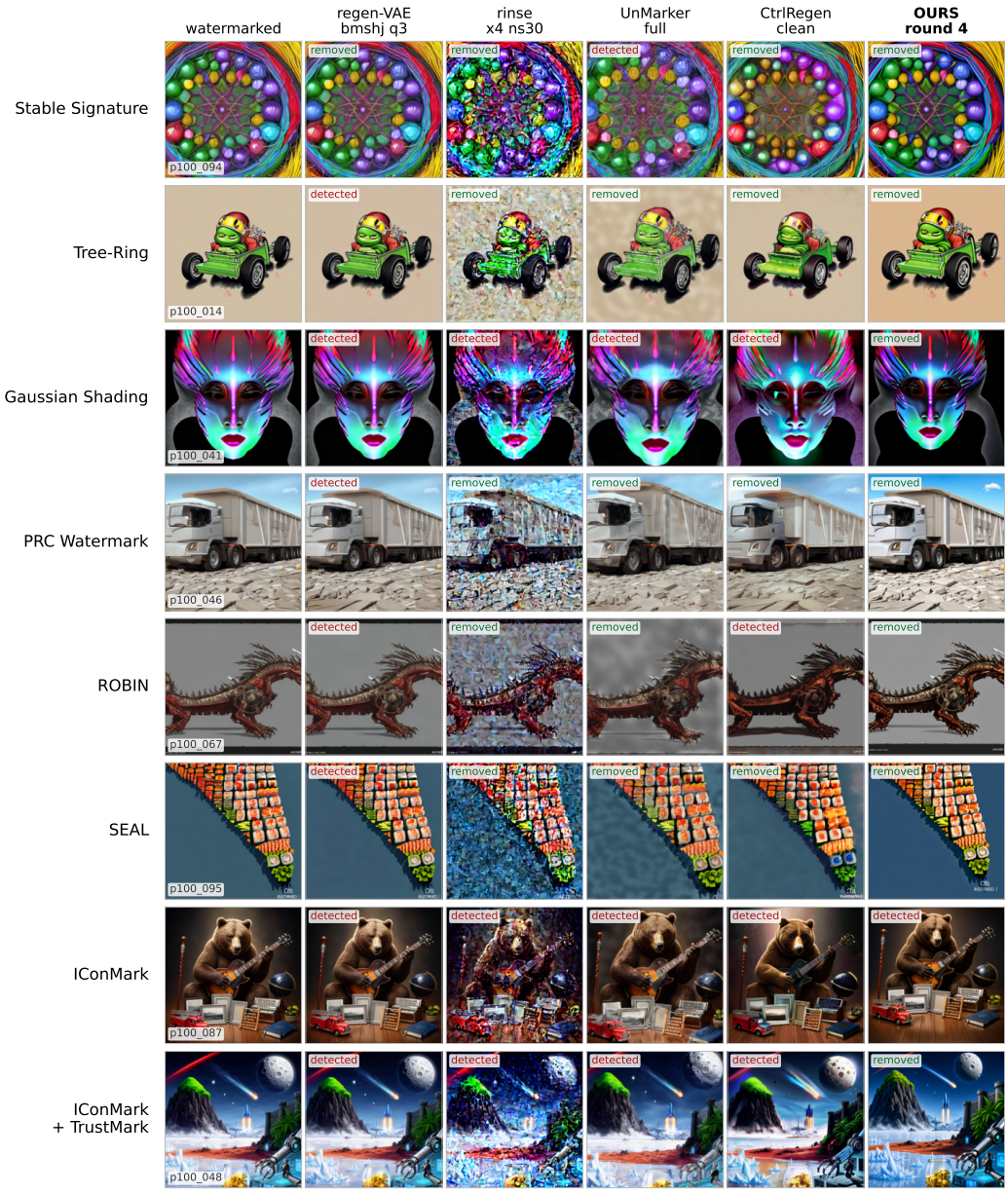

Figure 7: Qualitative comparison of watermark-removal attacks on six in-generation methods and two content-based variants. Each row shows a different watermarked input, outputs from four baseline attacks, and our GPT-Image-2 regeneration output after four rounds. Green "removed" labels indicate successful evasion of the watermark detector; red "detected" labels indicate that the watermark remains detectable under the evaluation thresholds.

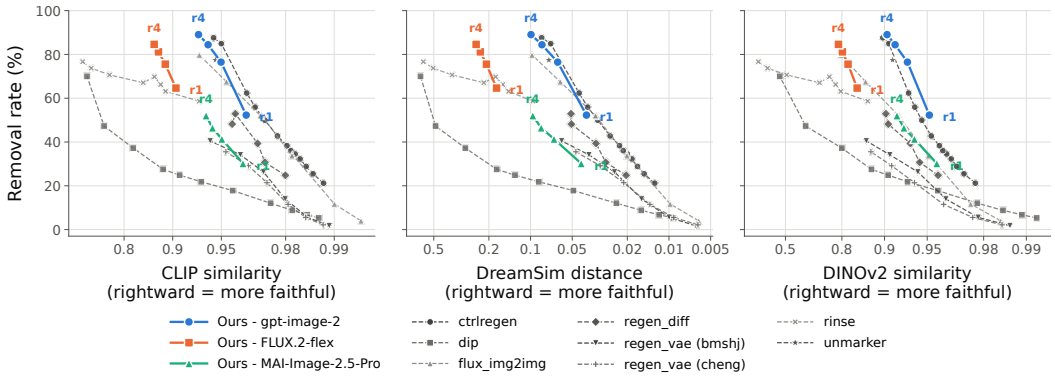

Figure 8: Removal-fidelity trade-offs averaged over 13 non-content-based watermarking methods. Our regeneration attack uses GPT-Image-2, FLUX.2-flex, and MAI-Image-2.5-Pro over four rounds, compared with baseline attacks at different strengths. Higher and farther right indicate better removal and fidelity, respectively.

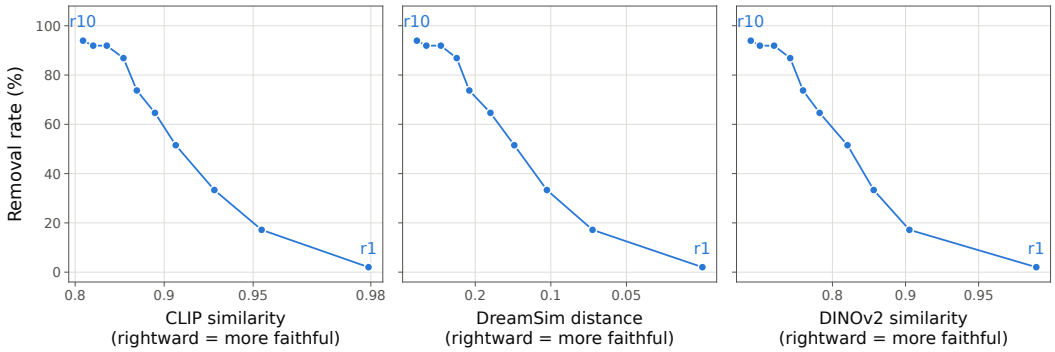

Figure 9: Removal-fidelity trade-offs for our attack on IConMark using extracted concepts. Points correspond to successive removal rounds, from R1 to R10. Fidelity is measured relative to the watermarked input. Higher and farther right indicate better removal and fidelity, respectively.

Table 4: Per-watermark removal rate (%, ↑) for nine baseline settings and three regeneration models. Our results use round 4. The denominator n contains only initially detected images. Refused regeneration sequences count as failures. Bold marks the highest removal rate in each row; it does not account for fidelity or computational cost. Baseline settings (column order): clean; full; $4\times$/ns30; $2\times$/ns60; c1/s0.15; ns60; q3; q3; it300. GPT: GPT-Image-2; FLUX: FLUX.2-flex; MAI: MAI-Image-2.5-Pro. Both IConMark variants use FLUX.2-flex for watermark generation.

		Baseline attacks								Ours (round 4)			
Watermark	n	Ctrl Regen	Un Marker	RINSE $4\times$	RINSE $2\times$	FLUX I2I	Diff. regen.	VAE Bmshj	VAE Cheng	DIP	GPT	FLUX	MAI
Post-hoc													
DwtDctSvd	96	**100.0**	99.0	99.0	96.9	66.7	72.9	80.2	74.0	51.0	96.9	99.0	96.9
RivaGAN	100	**100.0**	2.0	99.0	97.0	31.0	87.0	97.0	97.0	41.0	96.0	99.0	98.0
SSL WM	100	93.0	1.0	**99.0**	95.0	91.0	35.0	54.0	33.0	0.0	88.0	97.0	55.0
StegaStamp	100	**99.0**	59.0	80.0	35.0	0.0	1.0	0.0	0.0	0.0	85.0	83.0	32.0
TrustMark	100	**99.0**	78.0	98.0	96.0	89.0	44.0	0.0	1.0	0.0	91.0	96.0	65.0
VINE-R	100	90.0	**98.0**	1.0	0.0	0.0	0.0	0.0	0.0	0.0	**98.0**	70.0	60.0
ZoDiac	94	**98.9**	96.8	89.4	78.7	51.1	38.3	31.9	25.5	21.3	95.7	94.7	77.7
In-generation													
Gaussian Shading	100	0.0	**86.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**86.0**	31.0	23.0
PRC	100	98.0	**99.0**	23.0	7.0	31.0	0.0	1.0	6.0	0.0	**99.0**	98.0	45.0
ROBIN	100	86.0	98.0	**100.0**	**100.0**	0.0	0.0	0.0	0.0	0.0	75.0	83.0	33.0
SEAL	100	**100.0**	**100.0**	**100.0**	99.0	46.0	19.0	9.0	5.0	0.0	94.0	79.0	34.0
Stable Signature	100	**100.0**	92.0	**100.0**	**100.0**	34.0	**100.0**	67.0	35.0	1.0	80.0	99.0	27.0
Tree-Ring	100	75.0	**98.0**	30.0	16.0	7.0	2.0	2.0	2.0	0.0	73.0	71.0	30.0
Content-based													
IConMark	99	1.0	1.0	**10.1**	1.0	0.0	0.0	1.0	0.0	0.0	0.0	7.1	0.0
IConMark+TrustMark	100	2.0	2.0	12.0	2.0	1.0	1.0	0.0	0.0	0.0	3.0	**13.0**	1.0

Table 5: Per-watermark image fidelity for all baseline settings and our round-4 outputs, measured relative to the watermarked input. C: CLIP image similarity (↑); D: DreamSim distance (↓); V: DINOv2 cosine similarity (↑). Baseline scores use 100 images per configuration. Regeneration scores use the n_f images listed in Tables 7–9, excluding refused sequences and initially undetected images. Baseline settings (column order): clean; full; 4×/ns30; 2×/ns60; c1/s0.15; ns60; q3; q3; it300. GPT: GPT-Image-2; FLUX: FLUX.2-flex; MAI: MAI-Image-2.5-Pro. Both IConMark variants use FLUX.2-flex for watermark generation.

				Baseline attacks						Ours (round 4)			
Watermark	Metric	Ctrl Regen	Un Marker	RINSE 4×	RINSE 2×	FLUX I2I	Diff. regen.	VAE Bmshj	VAE Cheng	DIP	GPT	FLUX	MAI
Post-hoc													
DwtDctSvd	C	0.940	0.948	0.741	0.889	0.978	0.968	0.962	0.962	0.967	0.932	0.878	0.945
	D	0.090	0.071	0.371	0.138	0.021	0.031	0.029	0.025	0.024	0.098	0.217	0.081
	V	0.896	0.929	0.496	0.817	0.958	0.951	0.949	0.951	0.982	0.909	0.827	0.941
RivaGAN	C	0.938	0.948	0.744	0.892	0.980	0.973	0.963	0.965	0.968	0.932	0.889	0.947
	D	0.089	0.070	0.370	0.138	0.022	0.030	0.030	0.027	0.026	0.101	0.214	0.082
	V	0.897	0.927	0.508	0.820	0.953	0.950	0.948	0.949	0.979	0.910	0.832	0.941
SSL WM	C	0.943	0.945	0.730	0.880	0.977	0.969	0.969	0.969	0.973	0.927	0.883	0.945
	D	0.087	0.072	0.389	0.148	0.027	0.032	0.030	0.026	0.024	0.100	0.211	0.083
	V	0.893	0.923	0.485	0.802	0.943	0.946	0.947	0.945	0.982	0.904	0.828	0.933
StegaStamp	C	0.937	0.936	0.683	0.841	0.978	0.956	0.970	0.974	0.979	0.924	0.871	0.937
	D	0.090	0.077	0.445	0.189	0.025	0.045	0.030	0.024	0.019	0.110	0.226	0.087
	V	0.894	0.926	0.352	0.727	0.953	0.930	0.948	0.951	0.985	0.899	0.812	0.934
TrustMark	C	0.945	0.950	0.729	0.883	0.985	0.976	0.977	0.980	0.982	0.936	0.893	0.949
	D	0.083	0.069	0.377	0.140	0.020	0.029	0.025	0.019	0.017	0.091	0.202	0.074
	V	0.899	0.934	0.497	0.821	0.955	0.953	0.951	0.952	0.985	0.914	0.822	0.942
VINE-R	C	0.944	0.951	0.731	0.881	0.985	0.973	0.977	0.981	0.983	0.931	0.890	0.946
	D	0.083	0.070	0.375	0.141	0.019	0.030	0.024	0.019	0.017	0.096	0.205	0.078
	V	0.902	0.931	0.498	0.819	0.958	0.950	0.951	0.953	0.987	0.919	0.835	0.944
ZoDiac	C	0.945	0.948	0.726	0.874	0.987	0.973	0.975	0.981	0.984	0.936	0.883	0.947
	D	0.081	0.072	0.380	0.145	0.016	0.028	0.023	0.019	0.016	0.098	0.225	0.088
	V	0.901	0.921	0.510	0.813	0.963	0.950	0.945	0.941	0.984	0.915	0.824	0.937
In-generation													
Gaussian Shading	C	0.949	0.948	0.791	0.912	0.984	0.979	0.977	0.975	0.988	0.928	0.855	0.935
	D	0.078	0.074	0.336	0.135	0.017	0.023	0.021	0.019	0.010	0.097	0.286	0.108
	V	0.901	0.905	0.570	0.798	0.956	0.943	0.940	0.932	0.989	0.902	0.761	0.907
PRC	C	0.940	0.938	0.742	0.868	0.975	0.970	0.964	0.964	0.985	0.915	0.841	0.916
	D	0.084	0.081	0.414	0.179	0.028	0.033	0.032	0.029	0.013	0.126	0.294	0.136
	V	0.881	0.884	0.456	0.740	0.919	0.925	0.904	0.887	0.986	0.866	0.706	0.870
ROBIN	C	0.949	0.942	0.786	0.908	0.984	0.977	0.978	0.979	0.990	0.934	0.849	0.922
	D	0.081	0.080	0.341	0.133	0.017	0.023	0.019	0.017	0.009	0.099	0.302	0.130
	V	0.900	0.888	0.537	0.783	0.946	0.933	0.936	0.932	0.988	0.908	0.741	0.876
SEAL	C	0.944	0.943	0.805	0.910	0.984	0.978	0.977	0.977	0.989	0.939	0.860	0.934
	D	0.081	0.076	0.330	0.130	0.016	0.022	0.020	0.017	0.010	0.089	0.281	0.101
	V	0.901	0.901	0.570	0.810	0.955	0.946	0.941	0.934	0.988	0.906	0.756	0.909
Stable Signature	C	0.950	0.950	0.804	0.920	0.981	0.981	0.977	0.975	0.986	0.933	0.861	0.936
	D	0.075	0.069	0.321	0.123	0.017	0.022	0.018	0.017	0.011	0.092	0.269	0.100
	V	0.902	0.900	0.596	0.819	0.952	0.947	0.944	0.938	0.988	0.895	0.750	0.897
Tree-Ring	C	0.950	0.946	0.787	0.905	0.985	0.978	0.978	0.978	0.989	0.935	0.855	0.933
	D	0.080	0.078	0.344	0.136	0.016	0.023	0.020	0.018	0.010	0.093	0.284	0.105
	V	0.898	0.891	0.552	0.785	0.953	0.942	0.941	0.938	0.988	0.904	0.758	0.906
Content-based													
IConMark	C	0.919	0.939	0.751	0.889	0.975	0.963	0.967	0.971	0.982	0.939	0.869	0.964
	D	0.107	0.072	0.372	0.146	0.018	0.033	0.022	0.017	0.011	0.084	0.219	0.056
	V	0.864	0.905	0.487	0.776	0.956	0.932	0.937	0.938	0.990	0.907	0.809	0.947
IConMark+TrustMark	C	0.920	0.938	0.750	0.889	0.974	0.963	0.967	0.971	0.981	0.936	0.867	0.961
	D	0.107	0.071	0.373	0.146	0.019	0.033	0.022	0.018	0.012	0.087	0.222	0.059
	V	0.865	0.907	0.482	0.778	0.952	0.933	0.935	0.935	0.989	0.900	0.813	0.945

Table 6: Removal rate (%, ↑) across four regeneration rounds for all three models. Rates use the fixed initially detected population n. Refused sequences count as failures at every round, including earlier rounds that produced outputs; R1–R3 therefore reflect a four-round completion requirement. Both IConMark variants use FLUX.2-flex for watermark generation.

		GPT-Image-2				FLUX.2-flex				MAI-Image-2.5-Pro			
Watermark	n	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4
Post-hoc													
DwtDctSvd	96	96.9	97.9	97.9	96.9	97.9	100.0	100.0	99.0	97.9	96.9	99.0	96.9
RivaGAN	100	70.0	90.0	94.0	96.0	81.0	94.0	96.0	99.0	54.0	81.0	87.0	98.0
SSL WM	100	38.0	68.0	78.0	88.0	74.0	90.0	93.0	97.0	11.0	28.0	45.0	55.0
StegaStamp	100	25.0	48.0	70.0	85.0	50.0	65.0	79.0	83.0	10.0	23.0	25.0	32.0
TrustMark	100	42.0	81.0	87.0	91.0	83.0	90.0	95.0	96.0	33.0	50.0	56.0	65.0
VINE-R	100	78.0	94.0	97.0	98.0	46.0	59.0	69.0	70.0	28.0	41.0	50.0	60.0
ZoDiac	94	63.8	89.4	92.6	95.7	71.3	85.1	87.2	94.7	43.6	62.8	70.2	77.7
In-generation													
Gaussian Shading	100	44.0	70.0	81.0	86.0	22.0	23.0	29.0	31.0	17.0	19.0	22.0	23.0
PRC	100	73.0	93.0	96.0	99.0	71.0	86.0	91.0	98.0	31.0	37.0	40.0	45.0
ROBIN	100	24.0	56.0	68.0	75.0	55.0	77.0	79.0	83.0	18.0	24.0	27.0	33.0
SEAL	100	68.0	89.0	92.0	94.0	53.0	60.0	77.0	79.0	20.0	26.0	27.0	34.0
Stable Signature	100	34.0	66.0	73.0	80.0	88.0	98.0	99.0	99.0	13.0	21.0	27.0	27.0
Tree-Ring	100	23.0	52.0	71.0	73.0	48.0	55.0	61.0	71.0	16.0	28.0	29.0	30.0
Content-based													
IConMark	99	1.0	1.0	1.0	0.0	3.0	8.1	9.1	7.1	0.0	0.0	0.0	0.0
IConMark+TrustMark	100	1.0	1.0	2.0	3.0	4.0	7.0	14.0	13.0	1.0	1.0	1.0	1.0

Table 7: Round-by-round image fidelity for GPT-Image-2, measured relative to the watermarked input. CLIP and DINOv2 similarities are higher-is-better; DreamSim distance is lower-is-better. The fixed fidelity population n_f excludes initially undetected images and sequences refused at any round. Both IConMark variants use FLUX.2-flex for watermark generation.

		CLIP (↑)				DreamSim (↓)				DINOv2 (↑)			
Watermark	n_f	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4
Post-hoc													
DwtDctSvd	94	0.965	0.951	0.940	0.932	0.042	0.065	0.082	0.098	0.957	0.939	0.926	0.909
RivaGAN	98	0.964	0.950	0.941	0.932	0.046	0.069	0.087	0.101	0.955	0.935	0.922	0.910
SSL WM	98	0.962	0.947	0.935	0.927	0.046	0.068	0.086	0.100	0.950	0.928	0.914	0.904
StegaStamp	97	0.954	0.940	0.930	0.924	0.056	0.079	0.096	0.110	0.946	0.924	0.910	0.899
TrustMark	98	0.970	0.954	0.945	0.936	0.037	0.059	0.077	0.091	0.959	0.939	0.925	0.914
VINE-R	98	0.964	0.947	0.939	0.931	0.039	0.062	0.080	0.096	0.963	0.944	0.928	0.919
ZoDiac	92	0.968	0.953	0.945	0.936	0.037	0.063	0.081	0.098	0.961	0.941	0.926	0.915
In-generation													
Gaussian Shading	97	0.967	0.952	0.939	0.928	0.034	0.057	0.079	0.097	0.948	0.929	0.914	0.902
PRC	99	0.958	0.940	0.926	0.915	0.044	0.077	0.102	0.126	0.936	0.907	0.887	0.866
ROBIN	97	0.970	0.955	0.944	0.934	0.033	0.059	0.080	0.099	0.952	0.934	0.919	0.908
SEAL	96	0.968	0.955	0.945	0.939	0.032	0.055	0.073	0.089	0.953	0.933	0.918	0.906
Stable Signature	95	0.966	0.952	0.943	0.933	0.034	0.056	0.076	0.092	0.941	0.922	0.904	0.895
Tree-Ring	96	0.967	0.953	0.943	0.935	0.034	0.057	0.076	0.093	0.950	0.929	0.913	0.904
Content-based													
IConMark	98	0.970	0.957	0.947	0.939	0.031	0.051	0.068	0.084	0.956	0.936	0.920	0.907
IConMark+TrustMark	99	0.967	0.953	0.943	0.936	0.034	0.053	0.071	0.087	0.951	0.935	0.916	0.900

Table 8: Round-by-round image fidelity for FLUX.2-flex, measured relative to the watermarked input. CLIP and DINOv2 similarities are higher-is-better; DreamSim distance is lower-is-better. The fixed fidelity population n_f excludes initially undetected images and sequences refused at any round. Both IConMark variants use FLUX.2-flex for watermark generation.

		CLIP (↑)				DreamSim (↓)				DINOv2 (↑)			
Watermark	n_f	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4
Post-hoc													
DwtDctSvd	96	0.915	0.899	0.890	0.878	0.148	0.185	0.203	0.217	0.886	0.857	0.839	0.827
RivaGAN	100	0.917	0.903	0.894	0.889	0.151	0.178	0.202	0.214	0.883	0.857	0.841	0.832
SSL WM	100	0.919	0.903	0.891	0.883	0.137	0.174	0.194	0.211	0.881	0.851	0.839	0.828
StegaStamp	100	0.900	0.887	0.880	0.871	0.166	0.193	0.208	0.226	0.858	0.840	0.831	0.812
TrustMark	100	0.922	0.915	0.901	0.893	0.146	0.164	0.186	0.202	0.876	0.862	0.841	0.822
VINE-R	100	0.922	0.904	0.897	0.890	0.130	0.164	0.190	0.205	0.900	0.872	0.852	0.835
ZoDiac	94	0.914	0.904	0.893	0.883	0.161	0.188	0.212	0.225	0.880	0.852	0.837	0.824
In-generation													
Gaussian Shading	100	0.897	0.873	0.861	0.855	0.204	0.245	0.267	0.286	0.817	0.792	0.771	0.761
PRC	100	0.884	0.866	0.852	0.841	0.217	0.254	0.272	0.294	0.778	0.743	0.728	0.706
ROBIN	100	0.890	0.870	0.859	0.849	0.212	0.256	0.279	0.302	0.810	0.779	0.761	0.741
SEAL	100	0.898	0.880	0.865	0.860	0.207	0.239	0.262	0.281	0.813	0.786	0.765	0.756
Stable Signature	100	0.895	0.882	0.865	0.861	0.202	0.225	0.257	0.269	0.790	0.776	0.752	0.750
Tree-Ring	100	0.887	0.869	0.860	0.855	0.208	0.243	0.268	0.284	0.802	0.782	0.765	0.758
Content-based													
IConMark	99	0.910	0.889	0.879	0.869	0.147	0.182	0.201	0.219	0.874	0.844	0.830	0.809
IConMark+TrustMark	100	0.904	0.890	0.876	0.867	0.152	0.180	0.206	0.222	0.881	0.851	0.826	0.813

Table 9: Round-by-round image fidelity for MAI-Image-2.5-Pro, measured relative to the watermarked input. CLIP and DINOv2 similarities are higher-is-better; DreamSim distance is lower-is-better. The fixed fidelity population n_f excludes initially undetected images and sequences refused at any round. Both IConMark variants use FLUX.2-flex for watermark generation.

		CLIP (↑)				DreamSim (↓)				DINOv2 (↑)			
Watermark	n_f	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4
Post-hoc													
DwtDctSvd	95	0.967	0.955	0.950	0.945	0.037	0.057	0.070	0.081	0.973	0.959	0.949	0.941
RivaGAN	99	0.970	0.957	0.951	0.947	0.035	0.058	0.072	0.082	0.973	0.958	0.948	0.941
SSL WM	100	0.970	0.957	0.950	0.945	0.035	0.056	0.072	0.083	0.970	0.954	0.941	0.933
StegaStamp	100	0.959	0.948	0.941	0.937	0.043	0.064	0.078	0.087	0.965	0.951	0.941	0.934
TrustMark	100	0.972	0.961	0.955	0.949	0.030	0.050	0.064	0.074	0.973	0.961	0.950	0.942
VINE-R	100	0.968	0.957	0.951	0.946	0.031	0.053	0.066	0.078	0.978	0.961	0.953	0.944
ZoDiac	94	0.968	0.956	0.950	0.947	0.039	0.063	0.077	0.088	0.969	0.952	0.944	0.937
In-generation													
Gaussian Shading	91	0.963	0.949	0.941	0.935	0.045	0.073	0.093	0.108	0.952	0.929	0.916	0.907
PRC	91	0.948	0.934	0.924	0.916	0.069	0.099	0.120	0.136	0.923	0.897	0.883	0.870
ROBIN	84	0.951	0.933	0.928	0.922	0.065	0.095	0.115	0.130	0.921	0.897	0.885	0.876
SEAL	85	0.962	0.949	0.940	0.934	0.046	0.072	0.088	0.101	0.952	0.929	0.918	0.909
Stable Signature	86	0.962	0.949	0.941	0.936	0.042	0.069	0.087	0.100	0.943	0.920	0.907	0.897
Tree-Ring	87	0.963	0.949	0.941	0.933	0.044	0.070	0.089	0.105	0.952	0.932	0.917	0.906
Content-based													
IConMark	94	0.985	0.977	0.969	0.964	0.016	0.032	0.044	0.056	0.982	0.966	0.957	0.947
IConMark+TrustMark	95	0.985	0.974	0.967	0.961	0.017	0.034	0.047	0.059	0.981	0.966	0.954	0.945