SEALBENCH: A COMPREHENSIVE BENCHMARK FOR IMAGE WATERMARK DETECTION, ROBUSTNESS, AND RECOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

As concerns over the misuse of Generative AI models continue to grow, watermarking has emerged as a promising solution, embedding a detectable signal into generated or existing images. Existing image watermarking benchmarks have three limitations: 1) inconsistency across data for comparing in-generation and post-hoc watermarking methods, 2) uneven attack settings for image/watermark distortion, and 3) bit accuracy that does not measure whether the decoded message retains its meaning. To this end, we present SEALBENCH, an image watermarking benchmark that evaluates 4 in-generation and 11 post-hoc watermarking methods on 25,864 real, 15,000 manipulated, and 12,665 synthetic images. We further evaluate image-level watermarking attacks in four attack categories, from single image transformations to editing or watermark removal followed by additional processing, and propose the *attack-strength index (ASI)* to unify all image-level watermarking attacks on a single damage axis. We evaluate and propose *Question Success Rate (QSR)*, i.e. embedded message recovery, by testing whether a frozen language model can accurately answer questions using the extracted text. Our evaluation of *State-of-the-art* methods on SEALBENCH reveal interesting findings: a) both in-generation and post-hoc methods remain vulnerable to attacks, although their robustness varies significantly across attack types, b) attacks that produce similar image distortion can have different effects on watermark detection and c) High bit accuracy does not necessarily imply complete message recovery or preservation of the message's meaning. We believe SEALBENCH will support more consistent evaluation of image watermark robustness and encourage research on methods that preserve both watermark detection and message recovery under different attack settings.

1 INTRODUCTION

Generative image models can produce photorealistic images, increasing the need to verify their origin. Invisible watermarking provides one mechanism for tracing such content. A signal is embedded into an image which is later detected and decoded. Existing watermarking methods fall into two families. *In-generation* methods which embed the watermark during image generation, and *post-hoc* methods which embed it into an existing image. For both approaches, the watermark should remain detectable after compression, editing, and other image processing applied during distribution.

Existing benchmarks have three limitations. First, post-hoc and in-generation methods are often evaluated on different images and embedded messages. For example, WAVES (An et al., 2024) evaluates both approaches, but uses different images and message types for the methods. This makes it difficult to determine whether differences in performance arise from the watermarking methods or the data used to evaluate them. Second, attack parameters describe different operations: JPEG quality controls compression, rotation angle specifies the geometric transformation, and diffusion strength controls regeneration. Their values do not indicate how much each attack changes the image. Comparing watermark detection therefore requires measuring the resulting image distortion, rather than comparing parameter values alone. Third, bit accuracy does not establish whether the information in a decoded message remains recoverable (Christ & Gunn, 2024; Gunn et al., 2025;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

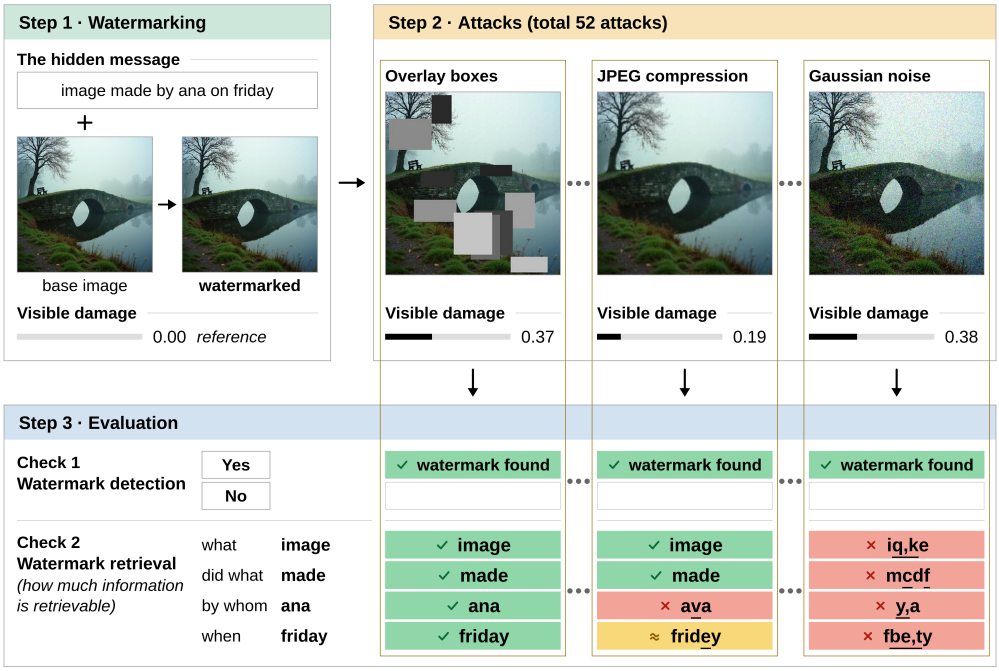

Figure 1: The SEALBENCH protocol for one image. **Step 1:** An English sentence is embedded as a watermark. **Step 2:** The watermarked image undergoes an attack; three of the 52 conditions are shown. ASI measures the LPIPS distance between the attacked and pre-attack images. **Step 3:** We evaluate watermark detection at a threshold calibrated using unwatermarked and attacked-unwatermarked images, and message recovery using questions answered from the decoded text. Detection succeeds in all three examples, but message recovery decreases, showing that detecting a watermark does not ensure recovery of its embedded information.

Gowal et al., 2025). A embedded message can retain most of its bits while its decoded message is no longer usable.

We introduce SEALBENCH, which evaluates 4 in-generation and 11 post-hoc methods on real, manipulated, and synthetic images (Figure 1; Table 1). On the synthetic evaluation, both post-hoc and in-generation methods watermark images from the same prompts and seeds and carry sentences from the same message corpus. We compare both in-generation and post-hoc watermarking methods on this *matched synthetic cohort*. Post-hoc methods are additionally evaluated across real and manipulated images.

The protocol combines three components. We measure attack-induced image distortion using ASI, defined as the LPIPS distance (Zhang et al., 2018) from the pre-attack image. The benchmark includes 52 attack conditions across four categories: single perturbations, model-based editing and watermark removal, multiple perturbations, and editing or removal followed by multiple perturbations. Finally, TPR@1%FPR measures detection, while Question Success Rate (QSR) measures recovery of information from sentence embedded messages. The benchmark records unwatermarked, watermarked, attacked-watermarked, and attacked-unwatermarked image states.

Our evaluation yields three findings. First, attacks with similar mean ASI produce different detection rates. On manipulated images, Gaussian noise and diffusion regeneration yield mean post-hoc TPR@1%FPR of 99.8% and 24.7% at mean ASI of 0.233 and 0.243, respectively. In-generation methods show the same pattern. Second, neither post-hoc nor in-generation watermarking performs better under every attack. On the matched synthetic cohort, in-generation methods achieve 68.4% mean TPR@1%FPR under category 2 model-based attacks, compared with 53.3% for post-hoc methods, but remain vulnerable to cropping and rotation. Third, high bit accuracy does not ensure message recovery. After attack, post-hoc methods on manipulated images achieve mean bit accuracy of 0.83, complete-message recovery of 0.39, and QSR of 0.44.

Table 1: Comparison with prior watermarking benchmarks. SEALBENCH evaluates both watermark families on a synthetic cohort matched by generator, prompt, and seed, compares attacks using a per-image damage axis, and evaluates message-level embedded message recovery.

	WAVES	W-Bench	SynthID-Image	SEALBENCH
Methods (in-gen + post-hoc)	2+3	0+13	0+8	4+11
Image categories	synth.+real	real	real	real+manipulated+synth.
Equal-damage comparison	per attack	no	no	ASI, per image
Semantic embedded messages	no	no	no	yes
Image states released	partial	partial	none	all four

Our contributions are:

- **Benchmark construction.** We evaluate 4 in-generation and 11 post-hoc methods under one protocol, using a synthetic cohort matched by generator, prompt, and seed. We provide four image states: unwatermarked, watermarked, attacked-watermarked, and attacked-unwatermarked.
- **Robustness evaluation.** We evaluate an extensive set of attack conditions across four categories, using ASI to compare attacks at similar image distortion and TPR@1%FPR to measure watermark detection. Detection thresholds are calibrated using unwatermarked and attacked-unwatermarked images.
- **Message recovery evaluation.** We introduce QSR to measure whether questions about an embedded message can be answered from its decoded text. We report QSR alongside bit accuracy and complete-message recovery.

2 RELATED WORK

Watermarking methods Watermarking methods are mainly divided into post-hoc and in-generation watermarking. Post-hoc methods embed messages into existing images, often using encoder-decoder networks trained against image distortions. Examples include StegaStamp (Tancik et al., 2020), MBRS (Jia et al., 2021), TrustMark (Bui et al., 2023), and VINE (Lu et al., 2025). In-generation methods embed watermarks during image generation by modifying sampling latents or the image decoder. Examples include Tree-Ring (Wen et al., 2023), RingID (Ci et al., 2024), Gaussian Shading (Yang et al., 2024), T2SMark (Yang et al., 2025) and PRC (Gunn et al., 2025). Table 2 lists the evaluated methods and their message capacities.

Watermarking Benchmarks. Classical benchmarks such as StirMark(Petitcolas et al., 1998; Petitcolas, 2000) compared watermarking schemes under standard attacks at a fixed visual quality. WAVES(An et al., 2024), the closest recent benchmark, standardized TPR at a low FPR and evaluated Tree-Ring, Stable Signature, and StegaStamp under distortion, regeneration, and adversarial attacks. However, it sets its headline thresholds on unwatermarked negatives and aggregates eight quality metrics per attack rather than scoring each image. W-Bench(Lu et al., 2025) evaluates post-hoc methods against image editing, and SynthID-Image(Gowal et al., 2025) reports the internal evaluation of a deployed system with transformed negatives and message-level code accuracy, but releases neither code nor data . These benchmarks do not explicitly evaluate sentence-level recovery or the readability of decoded messages. Table 1 summarizes these differences, and Appendix A describes the five principles behind the metric choices.

Attacks on Watermark. Regeneration attacks re-synthesize an image with a generative model and remove pixel-level watermarks (Zhao et al., 2024). Saberi et al. (2024) show that diffusion purification removes watermarks with small perturbations and that an adversarial attack on a substitute model removes those with larger ones, and UnMarker(Kassis & Hengartner, 2025) removes watermarks without detector queries or training data . Zhang et al. (2024a) prove that an attacker who preserves image quality can remove any strong watermark, so robustness is meaningful only relative to visible damage, which SEALBENCH measures with ASI.

3 SEALBENCH

We start the benchmarking effort by curating real, manipulated and synthetic images, followed by watermarking. We further incorporate an attack suite to evaluate the robustness of the proposed watermarking benchmark.

3.1 SOURCE IMAGES

Real and manipulated images. We use 25,864 real images and 15,000 manipulated images from MultiFakeVerse (Gupta et al., 2025). Real images come from EMOTIC (Kosti et al., 2020), PIC 2.0 (Liu et al., 2022), PIPA (Zhang et al., 2015), and PISC (Li et al., 2017). The manipulated images contain person-, object-, or scene-level edits to images from source images (Gupta et al., 2025).

Synthetic images. We generate 12,665 images via text prompt using SD2.1 (Rombach et al., 2022), SDXL(Podell et al., 2024), FLUX.1-dev(Black Forest Labs, 2024), Qwen-Image(Wu et al., 2025), HiDream-I1(Cai et al., 2025), and a distilled HunyuanImage-3.0 (Tencent Hunyuan Team, 2025) models. We use multiple generators to diversify the synthetic images and evaluate how watermark performance varies across generators (See Appendix E.1). Our text prompt description mainly contains *person-, object-, or scene-level* information. To get more fine-grained textual description, we further elaborate *person-, object-, or scene-level* information to eight different sub-classes: *single person's portrait, group of people, sports-scene, indoor-scene, food and drink, vehicle, landscape, or object*.

Together, these sources provide 25,864 real, 15,000 manipulated, and 12,665 synthetic images for SEALBENCH. We describe the watermarking methods applied to these image types next.

3.2 WATERMARKING

During the watermarking stage, we embed a hidden message in the image in form of bits. SEALBENCH includes *11 post-hoc* and *4 in-generation* watermarking methods. *Post-hoc* methods watermark on top of images such as VideoSeal(Fernandez et al., 2024), PixelSeal(Souček et al., 2025), TrustMark(Bui et al., 2023), WAM(Sander et al., 2025) etc. On the other hand, in-generation methods embed watermarks during generation process such as PRC(Gunn et al., 2025), Guassian Shading(Yang et al., 2024) etc. Based on the aforementioned type of watermarking strategy, we first create two versions of an image: 1) *Watermarked* image which contains either post-hoc or in-generation method (See Details in Table 2). 2) *Unwatermarked* image which does not contain any watermark. We further apply 52 different attack conditions on the watermarked and unwatermarked images mentioned next.

3.3 ATTACK SUITE

We apply 52 attack conditions organized into four attack categories:

1. Single Perturbation. Images in SEALBENCH contain 10 different image perturbations such as *JPEG compression, cropping, resizing, rotation, Gaussian blur, Gaussian noise, brightness, contrast, hue, and overlay*, each applied at three parameter settings. We measure how each watermark responds to the applied image perturbation. The complete perturbation parameter settings are reported in Appendix C Table 5.

2. 'Off-the-shelf' Model based Attacks. It contains 'off-the-shelf' model-based *image editing* and *watermark-removal* attacks. We use *prompt-based image editing* which changes the content of the image via Qwen-Image Edit(Wu et al., 2025) and FLUX.2-klein(Black Forest Labs, 2026). A vision-language model (VLMs) generates specific prompt to add, remove, or replace an object, person and/or scene. Further, an image editor (Qwen-Image-Edit(Wu et al., 2025) and FLUX.2-klein(Black Forest Labs, 2026)) edit the input image based on the prompt. To get the impact of VLMs on watermarking or image editing, we pass the same prompt to all image editor models. In addition to VLMs, we also apply VAE-based , diffusion based (Zhao et al., 2024), and UnMarker (Kassis & Hengartner, 2025) based image-to-image regeneration. These conditions are helpful in evaluating watermark robustness for model-based image editing, reconstruction and targeted watermark removal.

Table 2: Watermarking methods in SEALBENCH. Post-hoc methods watermark an existing image; in-generation methods embed the watermark during generation. Methods with fewer than 64 embedded message bits are not included in the sentence-level evaluation.

Post-hoc (11 methods)			*In-generation (4 methods; 12 generator-method pairs)*				
Method	Bits	Message	Method	embedded message	SD2.1	SDXL	FLUX.1
CIN (Ma et al., 2022)	30	N/A	RingID (Ci et al., 2024)	identity	✓	✓	✓
PIMoG (Fang et al., 2022)	30	N/A	Gauss. Shading (Yang et al., 2024)	256 bits	✓	✓	✓
WAM (Sander et al., 2025)	32	N/A	T2SMark (Yang et al., 2025)	256 bits	✓	✓	✓
TrustMark (Bui et al., 2023)	48	N/A	PRC (Gunn et al., 2025)	256 of 512	✓	✓	✓
EditGuard (Zhang et al., 2024b)	64	short					
VINE (Lu et al., 2025)	100	short					
InvisMark (Xu et al., 2025)	100	short					
OmniGuard (Zhang et al., 2025)	100	short					
MBRS (Jia et al., 2021)	256	long					
VideoSeal (Fernandez et al., 2024)	256	long					
PixelSeal (Souček et al., 2025)	256	long					

3. Multiple Perturbation. We apply 8 predefined sequences of image perturbations. In each sequence, the output of one perturbation becomes the input to the next. These sequences combine operations such as *resizing, cropping, filtering, overlay, and JPEG compression* to test watermark robustness after multiple processing steps. We label the sequences *messenger*, *filter*, *meme* etc. We want to highlight one sequence, i.e., *Resizing and compression*, which is widely used in social-media platforms (Giudice et al., 2017; Chuman et al., 2019).Appendix C Table 5 lists the sequences along with the parameter details.

4. Combining 'Off-the-shelf' Model-based Attacks and Multiple Perturbations. In this category, we apply 6 predefined sequences of operations, each starting with 'Off-the-shelf' Model-based Attacks followed by multiple perturbations. These sequences also test watermark detection and message recovery after the attacks. Appendix C Table 6 lists each sequence and its parameter settings.

3.4 SEALBENCH FINAL VERSION AND STATISTICS

The attack suite produces two additional image versions mentioned below (along with two versions highlighted in Sec. 3.2): 1) *Attacked-watermarked* image, which is obtained by applying an attack to a watermarked image; and 2) *Attacked-unwatermarked* image, which is obtained by applying the same attack condition to the corresponding unwatermarked image.

Overall, SEALBENCH contains 53,529 *unwatermarked images* and 739,818 *watermarked images*. We evaluate post-hoc watermarking methods on 1,600 real, 1,600 manipulated, and 1,593 synthetic images. These similarly sized subsets are used to compare post-hoc watermarking performance across the three image categories. The evaluation set contains 4,793 unwatermarked and 91,046 watermarked images, from which we produce 4,555,449 *attacked-watermarked* images and 241,826 distinct *attacked-unwatermarked* images. The attacked-watermarked and attacked-unwatermarked images are generated only for the evaluation-set images.

4 EVALUATION PROTOCOLS AND EVALUATION METRICS

The evaluation process of image watermarking begins with *bit retrieval* which was embedded in the image during the watermarking stage (See Section 3.2). Based on the retrieved bits, we further evaluate *A. watermark detection via Identity attribution*, or *B. watermark detection via message recovery*. In order to have a comprehensive analysis, we also evaluate *C. Attack-Strength Index and D. Robustness metrics*.

A. Watermark detection via Identity Attribution. Following Fernandez et al. (2023), we first create a fixed set of identity codebook. Our ablation study suggest that a codebook with 100 identities is sufficient for the benchmark (See Appendix E.3). After codebook formation, we compare the recovered bits from image with each candidate identity codeword mentioned in the codebook. The bit matching score decide two important factors:

1. If the bit matching score crosses a pre-defined threshold, then the image is categorized into *watermarked* image; otherwise, it is categorized as *unwatermarked*. Following An et al. (2024); Lu et al. (2025), we evaluate watermark detection using **TPR@1%FPR**.

2. After watermark detection, the bit matching score also decide with which identity it has highest overlap. Thus, it retrieve identity attribute from a watermarked image. Following An et al. (2024); Fernandez et al. (2023), we report **open-set attribution accuracy (in @1%FPR)** as the fraction of all evaluated watermarked images for which the watermark is successfully detected and assigned to the correct embedded identity.

B. Watermark detection via Message Recovery. Following Lu et al. (2025), we compare the recovered bits directly with a known embedded message from SEALBENCH's ground truth metadata. The bit matching score decides two important factors:

1. If the bit matching score crosses a pre-defined threshold, then the image is categorized into *watermarked* image; otherwise, it is categorized as *unwatermarked*. Following An et al. (2024); Lu et al. (2025), we evaluate watermark detection using **TPR@1%FPR**.

2. From the bit matching score, we first calculate **bit accuracy (in %)** i.e. the fraction of embedded message bits decoded correctly. Then, *we convert the recovered bits into text using the same 5-bit-per-character encoding used to embed the sentence*. Based on the decoded sentence, we evaluate **complete-message recovery**, and **Question Success Rate (QSR)**.

2.1. ***Complete-message recovery:*** is evaluated in terms of **accuracy (in %)** of recovered messages whose decoded sentences exactly match the ground truth embedded sentences Gowal et al. (2025).

2.2. ***Question Success Rate (QSR):*** We propose this metric to address the gap of partial sentence recovery. A sentence that is not recovered completely may still contain information from the original message. We measure this using Question Success Rate (QSR). A frozen Qwen2.5-7B-Instruct(Qwen Team, 2024) *reader* receives the recovered text and answers stored questions about the original message, without access to the original sentence. QSR is the fraction of questions answered correctly according to the scoring rules mentioned in Appendix D.2. We assess sensitivity to *reader* choice using Qwen3.5-9B(Qwen Team, 2026), Ministral-8B-Instruct-2410(Mistral AI, 2024), Gemma-3-12B-IT(Gemma Team, 2025), and Llama-3.1-8B-Instruct(Grattafiori et al., 2024). All readers have same hyperparameter setup; i.e. frozen weights and use the same inputs, decoding limits, and scoring % rules. Qwen2.5-7B-Instruct is the primary reader, and Appendix F.5 reports the sensitivity results in detail.

C. Attack-Strength Index (ASI). To evaluate the robustness of the watermarks over attack suits (See Section 3.3, we propose Attack-Strength Index (ASI). As attack parameters, such as JPEG quality, rotation angle, and diffusion strength, use different scales and cannot be compared directly, we therefore define ASI as the LPIPS distance (Zhang et al., 2018) between an attacked image and its corresponding pre-attack or original image. ASI is computed separately for each image in SEALBENCH. We compare detection and recovery of image at similar mean ASI to assess robustness at comparable image distortion.

D. Robustness metrics. We evaluate detection robustness as a function of ASI. For each watermarking method, image category, and attack type, each evaluated condition defines one point on a 2D plane, with mean ASI on the x-axis and **TPR@1%FPR** on the y-axis. The attack conditions used in this evaluation are listed in Appendix C. We summarize these curves using two metrics. *AURSC* is the normalized area under the **TPR@1%FPR-versus-ASI curve** over the range $[0, 0.2]$. It is computed separately for each attack type and then averaged equally across all attack types. A higher AURSC indicates that detection remains stronger over this ASI range. *bit-AURSC* uses the same calculation with bit accuracy on the y-axis and mean ASI on the x-axis. *sem-AURSC* uses QSR on the y-axis and mean ASI on the x-axis. *Observed Min@0.2* is the lowest **TPR@1%FPR** directly observed among evaluated attack conditions whose mean ASI is at most 0.2. It therefore reports the lowest measured detection performance within the same ASI range. Both metrics lie in $[0, 1]$. Appendix E.4 provides the formal definitions and curve-construction procedure.

Comparison protocol. Comparison between in-generation and post-hoc watermarking methods use the matched synthetic cohort of 1,193 images, with TPR@1%FPR for detection and QSR for methods that support sentence embedded messages. Post-hoc methods are further evaluated on 1,600

Table 3: Open-set attribution and watermark detection on the 1,193-image synthetic cohort matched by generator, prompt, and seed. The identity attribution block reports open-set attribution accuracy (in @1%FPR) budget, using 100 candidate identities for the 11 post-hoc methods and 1,024 native keys for RingID. The message recovery block reports TPR@1%FPR (%) for seven post-hoc and three in-generation methods, using bit agreement with the known embedded sentence payload as the detection score. Unattacked denotes performance before attack, C1–C4 denote attack categories, and All is the mean over attacked conditions.

Post-Hoc or In-Gen	Method	Unattacked	C1	C2	C3	C4	All
Identity attribution (open-set attribution; RingID on its 1,024 keys)							
in-gen.	RingID	100.0	82.2	79.0	81.2	64.4	80.1
post-hoc	CIN	100.0	70.2	13.6	19.1	0.4	46.9
post-hoc	PIMoG	100.0	72.2	42.2	49.0	25.2	59.5
post-hoc	WAM	100.0	97.8	63.8	99.9	66.6	**90.2**
post-hoc	TrustMark	99.9	77.6	22.7	83.6	25.7	65.5
post-hoc	EditGuard	100.0	61.1	7.6	29.6	1.2	42.4
post-hoc	VINE	100.0	80.2	**95.4**	75.3	**74.4**	81.0
post-hoc	InvisMark	100.0	93.3	19.6	87.4	17.2	74.2
post-hoc	OmniGuard	100.0	82.3	78.5	87.5	69.4	81.3
post-hoc	MBRS	100.0	79.8	39.8	75.1	51.3	70.5
post-hoc	VideoSeal	100.0	94.6	55.7	**100.0**	59.9	86.4
post-hoc	PixelSeal	100.0	**99.7**	43.2	**100.0**	57.7	87.5
Message Recovery							
in-gen.	Gauss. Shading	100.0	81.0	94.5	76.3	67.7	80.7
in-gen.	T2SMark	100.0	80.1	75.9	75.2	52.7	76.6
in-gen.	PRC	100.0	74.7	34.8	61.0	24.1	63.6
post-hoc	EditGuard	100.0	63.1	12.7	45.8	8.0	47.7
post-hoc	VINE	100.0	80.1	**97.4**	75.2	76.6	81.4
post-hoc	InvisMark	100.0	89.8	11.8	72.0	7.5	67.7
post-hoc	OmniGuard	100.0	83.1	88.5	87.5	**78.1**	84.0
post-hoc	MBRS	100.0	79.6	45.7	75.0	52.2	71.3
post-hoc	VideoSeal	100.0	96.2	68.2	100.0	64.9	**89.7**
post-hoc	PixelSeal	100.0	**99.8**	48.9	**100.0**	60.7	88.7

real, 1,600 manipulated, and 1,593 synthetic images using the same metrics, together with open-set attribution accuracy (in @1%FPR)

5 RESULTS

5.1 OVERALL DETECTION PERFORMANCE

Before attack, all 11 post-hoc methods achieve at least 99.9% TPR@1%FPR in the identity attribution setting across the three image categories (Appendix Table 9). On the matched synthetic cohort, RingID achieves 100% in the identity attribution setting, and the three in-generation methods that embed sentences achieve 100% in the message recovery setting (Table 3).

After attack, the same eight post-hoc methods fall below 7% TPR@1%FPR on an attack with mean ASI ≤ 0.2 in each image category (Appendix Table 9, Obs. Min@0.2 columns). Thus, high detection before attack does not establish robustness after attack, even within the specified ASI budget.

5.2 DETECTION ACROSS ATTACK TYPES

We compare attacks at similar mean ASI to assess their effects on different watermark methods at comparable image distortion. On manipulated images, Gaussian noise ($\sigma = 10$) and diffusion regeneration at strength 0.2 produce mean ASI of 0.233 and 0.243, respectively. We observe that mean post-hoc TPR@1%FPR is 99.8% under noise but 24.7% under diffusion regeneration. Thus, diffusion regeneration reduces detection more than Gaussian noise despite producing similar measured distortion. We observe the same pattern for in-generation methods. Gaussian noise ($\sigma = 5$) and diffusion regeneration at strength 0.2 produce mean ASI of 0.161 and 0.154, respectively, while mean TPR@1%FPR is 98.5% and 74.7%. The difference between these attacks therefore occurs in both watermark families.

These results show that attacks causing similar image distortion can have different effects on watermark detection. Comparing detection rates at similar mean ASI identifies attacks that cause larger reductions in watermark detection at comparable image distortion.

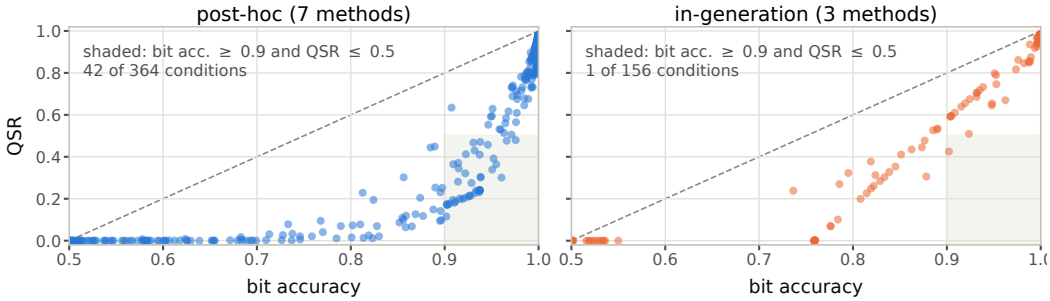

Figure 2: Bit accuracy against QSR for each method-condition pair. Left: 7 post-hoc methods on the matched synthetic cohort. Right: 3 in-generation methods on the same cohort. The shaded region marks cases with bit accuracy ≥ 0.9 and QSR ≤ 0.5.

5.3 Comparison between Post-Hoc and In-Generation Watermarking

Tables 3 and 4 compare detection and message recovery on the matched synthetic cohort in the message recovery setting. Across attacked conditions, mean TPR@1%FPR is 75.8% for post-hoc methods and 73.6% for in-generation methods, with method-level averages ranging from 47.7-89.7% and 63.6-80.7%, respectively. Mean QSR is 0.454 (range 0.104-0.629) for post-hoc methods and 0.635 (range 0.527-0.693) for in-generation methods. Thus, similar average detection accompanies different message-recovery performance.

Performance also varies across attacks. In-generation methods fall to 0-3% TPR@1%FPR under strong cropping or rotation. Under model-based attacks in category 2, however, they achieve higher mean detection than post-hoc methods (68.4% versus 53.3%) and higher mean QSR (0.354 versus 0.154). Under multiple perturbations in category 3, VideoSeal and PixelSeal achieve QSR of 0.822 and 0.811, exceeding the highest in-generation value of 0.727.

Both post-hoc and in-generation watermarking therefore remain vulnerable to attacks. In-generation methods perform better on average under model-based attacks, while some post-hoc methods achieve higher message recovery under multiple perturbations. Overall averages should therefore be interpreted alongside results for individual methods and attack types.

5.4 Performance across generators

We compare in-generation methods across their supported generators (Appendix F.4 Table 14). Gaussian Shading achieves mean TPR@1%FPR of 82.2%, 80.3%, and 79.6% on SD2.1, SDXL, and FLUX.1, respectively. Its detection rate therefore remains similar across the three generators, with a difference of 2.6 percentage points between the highest and lowest values. In contrast, PRC achieves 76.8%, 66.6%, and 47.6%, respectively, a difference of 29.2 percentage points. These results show that a watermarking method can perform differently depending on the generator used. Robustness should therefore be evaluated on each supported generator, rather than inferred from results on a single generator.

5.5 Embedded message recovery

We evaluate bit accuracy, complete-message recovery, and QSR for the seven post-hoc and three in-generation methods that carry sentence messages. On the matched synthetic cohort under attack, post-hoc methods have mean bit accuracy of 0.835, complete-message recovery of 0.396, and QSR of 0.454. The corresponding values for in-generation methods are 0.883, 0.610, and 0.635 (Appendix Table 19). These metrics measure different outcomes: correctly decoded bits, exact sentence recovery, and correct answers to questions about the embedded message.

High bit accuracy does not always correspond to high QSR (Figure 2). On the matched cohort, 42 of 364 post-hoc method-condition pairs and 1 of 156 in-generation pairs have bit accuracy ≥ 0.9 but QSR ≤ 0.5. The same discrepancy remains when both metrics are summarized over the same ASI range. For example, PixelSeal achieves a bit-AURSC of 0.958 but a sem-AURSC of 0.821 (Table 4). These results show that high bit accuracy does not necessarily imply preservation of the information measured by the stored questions.

Table 4: Message recovery on the matched synthetic cohort (Section 4). QSR is reported before attack (Unattacked), for each attack category (C1–C4), and across all attacked conditions (All). Class denotes message length. Msg. Rec. and Bit Acc. report mean complete-message recovery and bit accuracy across attacked conditions. sem-AURSC and bit-AURSC summarize the corresponding QSR and bit-accuracy curves over ASI $[0, 0.2]$, averaged across all attack types. Appendix Table 18 reports post-hoc results on real, manipulated, and all 1,593 synthetic images. Table 18.

		QSR									
Method	Class	Unattacked	C1	C2	C3	C4	All	sem-AURSC	bit-AURSC	Msg. Rec.	Bit Acc.
post-hoc, matched synthetic cohort											
EditGuard	short	0.868	0.388	0.002	0.178	0.000	0.261	0.574	0.851	0.285	0.673
VINE	short	0.859	0.679	0.403	0.636	0.388	0.604	0.768	0.966	0.663	0.894
InvisMark	short	0.240	0.155	0.005	0.065	0.000	0.104	0.172	0.863	0.021	0.781
OmniGuard	short	0.868	0.646	0.310	0.701	0.330	0.575	0.746	0.959	0.618	0.894
MBRS	long	0.982	0.528	0.122	0.307	0.066	0.389	0.674	0.903	0.247	0.796
VideoSeal	long	0.976	0.758	0.123	0.822	0.121	0.614	0.828	0.962	0.457	0.903
PixelSeal	long	0.976	0.794	0.112	0.811	0.090	0.629	0.821	0.958	0.484	0.907
in-generation, matched synthetic cohort											
Gauss. Shading	long	0.984	0.776	0.457	0.706	0.308	0.685	0.877	0.965	0.629	0.886
T2SMark	long	0.984	0.779	0.455	0.727	0.295	0.693	0.882	0.960	0.665	0.875
PRC	long	0.983	0.657	0.149	0.508	0.106	0.527	0.802	0.955	0.536	0.888

Complete-message recovery and QSR also provide different information. In the partial-recovery analysis (Appendix Table 20), 27.3% of correct answers for post-hoc methods come from decoded messages that do not exactly match the embedded sentence, compared with 5.1% for in-generation methods. Some questions can therefore be answered despite errors in the decoded sentence. Conversely, exact recovery does not guarantee a correct answer because the reader can also make errors.

Across five readers, mean attacked QSR on the matched cohort ranges from 0.427 to 0.507 for post-hoc methods and from 0.623 to 0.647 for in-generation methods (Appendix Table 21). Condition-level QSR has Pearson's $r \geq 0.985$ between every pair of readers. The in-generation method ranking is unchanged, but the ranking of the four highest-scoring post-hoc methods varies with the reader. Thus, readers give closely correlated condition-level scores, while absolute QSR and some method rankings remain reader-dependent. QSR should therefore be interpreted alongside complete-message recovery and the reader sensitivity results (Appendix F.5).

Summary of results. Both post-hoc and in-generation watermarking remain vulnerable to attacks, with performance varying across watermarking methods, attack types, and generators. Attacks with similar mean ASI can produce different detection rates, supporting comparisons that consider both attack type and image distortion. High bit accuracy does not ensure complete-message recovery, while partially recovered messages can still support correct answers to stored questions. These results support reporting detection, bit accuracy, complete-message recovery, and QSR together to characterize watermark robustness.

6 Conclusion

SEALBench establishes a common evaluation protocol for 4 in-generation and 11 post-hoc image watermarking methods under 52 attack conditions. The benchmark measures watermark detection, identity attribution, and message recovery before and after attack. Post-hoc and in-generation methods are compared on synthetic images matched by generator, prompt, and seed, while post-hoc methods are additionally evaluated on real and manipulated images. ASI enables comparisons of watermark detection and message recovery at similar levels of image distortion. We release the image states, attack specifications, and evaluation tools to support reproducible comparisons and future research on watermark robustness.

Limitations and future work. Our results are limited to the evaluated methods, generators, and attack conditions. In-generation methods are evaluated on three of the six generators, and robustness experiments use subsets of 1,600 real, 1,600 manipulated, and 1,593 synthetic images. The robustness summaries also depend on interpolation and constant extension where measurements do not cover the reporting range. Message recovery is evaluated only for methods with sufficient sentence capacity, and QSR depends on the stored questions and the reader, despite sensitivity analysis with four additional readers. Future work can extend generator and attack coverage, and compare distortion metrics and message-recovery tasks.

ETHICS STATEMENT

SEALBENCH uses real and manipulated images from MultiFakeVerse (Gupta et al., 2025); no new photographs were collected. Source images remain subject to their original licences and terms of use, including the applicable MultiFakeVerse access agreement. Because these images may depict identifiable individuals, the manipulated images are used as research examples and should not be interpreted as records of actual events. The synthetic images are generated from prompts describing fictional people, objects, and scenes and are not intended to depict real individuals.

Our watermark-removal experiments assess the limitations of existing methods and support the development of more robust watermarking. Although the findings could be misused to evade detection, evaluating these vulnerabilities is important for understanding when watermark-based provenance verification can be relied upon.

REPRODUCIBILITY STATEMENT

All generation, attack, and evaluation procedures are seeded, and the corresponding commands are released with the toolchain (Appendix G). The benchmark releases the prompt corpus, embedded message files, per-image metadata, unwatermarked and watermarked images, frozen chain specification, evaluation subsets, and calibration thresholds with their per-stratum false-positive rates. Scripts are provided to generate the corresponding attacked-unwatermarked and attacked-watermarked images. We distinguish between reproducing the reported results and rerunning the watermarking pipeline from scratch. The released per-image records and metric files are sufficient to reproduce every table and figure in this paper, which are generated from these files by a single script. Re-running watermark embedding and decoding additionally requires the method keys and identity codebooks. These are not included in the public release, so the watermarked images cannot be regenerated from the released files alone; we provide the keys and codebooks to researchers on request for benchmark use.

Images derived from MultiFakeVerse are released only to the extent permitted by its licence (see the ethics statement). The synthetic images, embedded messages, attack specifications, evaluation records, and code are independent of this licence.

AI USE STATEMENT

Generative image models were used to create the synthetic images and to perform model-based editing and regeneration attacks. A vision-language model generated image-specific instructions for the prompt-based editing attacks, as described in Section 3. Frozen language models were also used to answer questions from decoded messages for QSR evaluation and reader sensitivity analysis, as described in Section 4 and Appendix F.5.

AI tools assisted with manuscript drafting, language editing, and LaTeX formatting. The authors are responsible for the final manuscript, experimental methodology, and reported results.

REFERENCES

Bang An, Mucong Ding, Tahseen Rabbani, Aakriti Agrawal, Yuancheng Xu, Chenghao Deng, Sicheng Zhu, Abdirisak Mohamed, Yuxin Wen, Tom Goldstein, and Furong Huang. WAVES: Benchmarking the robustness of image watermarks. In *International Conference on Machine Learning (ICML)*, 2024.

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

Black Forest Labs. FLUX.2 [klein]: Towards interactive visual intelligence. `https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence`, January 2026.

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

Qi Cai et al. HiDream-I1: A high-efficient image generative foundation model with sparse diffusion transformer. arXiv:2505.22705, 2025.

Miranda Christ and Sam Gunn. Pseudorandom error-correcting codes. In *Advances in Cryptology – CRYPTO 2024, LNCS 14925*, pp. 325–347. Springer, 2024.

Tatsuya Chuman, Kenta Iida, Warit Sirichotedumrong, and Hitoshi Kiya. Image manipulation specifications on social networking services for encryption-then-compression systems. *IEICE Transactions on Information and Systems*, E102.D(1):11–18, 2019.

Hai Ci, Pei Yang, Yiren Song, and Mike Zheng Shou. RingID: Rethinking Tree-Ring watermarking for enhanced multi-key identification. In *European Conference on Computer Vision (ECCV)*, 2024.

Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, second edition, 2007.

Han Fang, Zhaoyang Jia, Zehua Ma, Ee-Chien Chang, and Weiming Zhang. PIMoG: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network. In *ACM International Conference on Multimedia (ACM MM)*, pp. 2267–2275, 2022.

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

Pierre Fernandez, Hady Elsahar, I. Zeki Yalniz, and Alexandre Mourachko. Video Seal: Open and efficient video watermarking. *arXiv preprint arXiv:2412.09492*, 2024.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Oliver Giudice, Antonino Paratore, Marco Moltisanti, and Sebastiano Battiato. A classification engine for image ballistics of social data. In *Image Analysis and Processing (ICIAP), LNCS*, pp. 625–636. Springer, 2017.

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

Parul Gupta, Shreya Ghosh, Tom Gedeon, Thanh-Toan Do, and Abhinav Dhall. Multiverse through deepfakes: The MultiFakeVerse dataset of person-centric visual and conceptual manipulations. In *ACM International Conference on Multimedia (ACM MM)*, pp. 13258–13265, 2025.

Zhaoyang Jia, Han Fang, and Weiming Zhang. MBRS: Enhancing robustness of DNN-based watermarking by mini-batch of real and simulated JPEG compression. In *ACM International Conference on Multimedia (ACM MM)*, 2021.

Simon Josefsson. The Base16, Base32, and Base64 data encodings. RFC 4648, Internet Engineering Task Force, October 2006.

Andre Kassis and Urs Hengartner. UnMarker: A universal attack on defensive image watermarking. In *IEEE Symposium on Security and Privacy (S&P)*, 2025.

Ronak Kosti, Jose M. Alvarez, Adria Recasens, and Agata Lapedriza. Context based emotion recognition using EMOTIC dataset. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(11):2755–2766, 2020.

Martin Kutter and Fabien A. P. Petitcolas. A fair benchmark for image watermarking systems. In *Security and Watermarking of Multimedia Contents, Proc. SPIE Electronic Imaging '99, vol. 3657*, pp. 226–239, 1999.

Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S. Kankanhalli. Dual-Glance model for deciphering social relationships. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

Si Liu, Zitian Wang, Yulu Gao, Lejian Ren, Yue Liao, Guanghui Ren, Bo Li, and Shuicheng Yan. Human-centric relation segmentation: Dataset and solution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4987–5001, 2022.

Shilin Lu, Zihan Zhou, Jiayou Lu, Yuanzhi Zhu, and Adams Wai-Kin Kong. Robust watermarking using generative priors against image editing: From benchmarking to advances. In *International Conference on Learning Representations (ICLR)*, 2025.

Rui Ma, Mengxi Guo, Yi Hou, Fan Yang, Yuan Li, Huizhu Jia, and Xiaodong Xie. Towards blind watermarking: Combining invertible and non-invertible mechanisms. In *ACM International Conference on Multimedia (ACM MM)*, 2022.

Mistral AI. Un Ministral, des Ministraux. `https://mistral.ai/news/ministraux`, October 2024.

Fabien A. P. Petitcolas. Watermarking schemes evaluation. *IEEE Signal Processing Magazine*, 17 (5):58–64, 2000.

Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Attacks on copyright marking systems. In *Second International Workshop on Information Hiding (IH'98), LNCS 1525*, pp. 218–238. Springer, 1998.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations (ICLR)*, 2024.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Qwen Team. Qwen3.5: Towards native multimodal agents. `https://qwen.ai/blog?id=qwen3.5`, February 2026.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

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 (ICLR)*, 2024.

Tom Sander, Pierre Fernandez, Alain Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages. In *International Conference on Learning Representations (ICLR)*, 2025.

Tomáš Souček, Pierre Fernandez, Hady Elsahar, Sylvestre-Alvise Rebuffi, Valeriu Lacatusu, Tuan Tran, Tom Sander, and Alexandre Mourachko. Pixel Seal: Adversarial-only training for invisible image and video watermarking. *arXiv preprint arXiv:2512.16874*, 2025.

Matthew Tancik, Ben Mildenhall, and Ren Ng. StegaStamp: Invisible hyperlinks in physical photographs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2126, 2020.

Tencent Hunyuan Team. HunyuanImage 3.0 technical report. arXiv:2509.23951, 2025.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-Ring watermarks: Fingerprints for diffusion images that are invisible and robust. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Chenfei Wu et al. Qwen-Image technical report. arXiv:2508.02324, 2025.

Rui Xu, Mengya Hu, Deren Lei, Yaxi Li, David Lowe, Alex Gorevski, Mingyu Wang, Emily Ching, and Alex Deng. InvisMark: Invisible and robust watermarking for AI-generated image provenance. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

Jindong Yang, Han Fang, Weiming Zhang, Nenghai Yu, and Kejiang Chen. T2SMark: Balancing robustness and diversity in noise-as-watermark for diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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

Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the sand: Impossibility of strong watermarking for generative models. In *International Conference on Machine Learning (ICML)*, 2024a.

Ning Zhang, Manohar Paluri, Yaniv Taigman, Rob Fergus, and Lubomir Bourdev. Beyond frontal faces: Improving person recognition using multiple cues. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4804–4813, 2015.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Xuanyu Zhang, Runyi Li, Jiwen Yu, Youmin Xu, Weiqi Li, and Jian Zhang. EditGuard: Versatile image watermarking for tamper localization and copyright protection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11964–11974, 2024b.

Xuanyu Zhang, Zecheng Tang, Zhipei Xu, Runyi Li, Youmin Xu, Bin Chen, Feng Gao, and Jian Zhang. OmniGuard: Hybrid manipulation localization via augmented versatile deep image watermarking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative AI. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

A Evaluation principles

We organize the evaluation around five principles informed by prior watermarking research. These principles guide the choice of metrics and reporting protocol; they are not additional cryptographic guarantees.

R1: Task-specific performance. Detection, attribution, and embedded message recovery measure different outcomes and should be evaluated separately (Cox et al., 2007; An et al., 2024; Gowal et al., 2025). We report detection using TPR@1%FPR and attribution using open-set accuracy at the calibrated 1% FPR operating point. embedded message recovery is evaluated using bit accuracy, complete-message recovery, and QSR (Section 4).

R2: False-positive calibration and baseline evaluation. False-positive control distinguishes watermark detection from acceptance of unwatermarked inputs (Gunn et al., 2025; An et al., 2024). We calibrate detection thresholds using unwatermarked and attacked-unwatermarked negatives, with an empirical pooled FPR of at most 1%. This target applies to the calibration pool rather than to every attack condition separately.

R3: Robustness within a specified damage range. Watermark detection after attack should be interpreted together with the resulting image change (An et al., 2024). We report detection as a function of ASI and summarize it over $[0, 0.2]$ using AURSC and observed Min@0.2. AURSC uses interpolated curves, whereas observed Min@0.2 includes only measured conditions within the reporting range. The observed minimum does not bound performance at unmeasured conditions.

R4: Comparison at similar image distortion. Attack parameters are defined on different scales, so their values do not provide a common measure of image distortion (Petitcolas et al., 1998; Kutter & Petitcolas, 1999; Petitcolas, 2000; An et al., 2024). We measure ASI for each attacked image and compare detection at similar condition-level mean ASI. Detection performance and image distortion

are reported separately. Similar mean ASI does not imply identical per-image damage distributions or identical perceptual effects.

R5: Message-level recovery. Recovery of an embedded message is a distinct objective from correctly decoding individual bits (Christ & Gunn, 2024; Gunn et al., 2025). We therefore report complete-message recovery alongside bit accuracy. For sentence embedded messages, QSR additionally measures whether a reader can answer stored questions using the decoded text. These metrics distinguish exact sentence recovery from recovery of the information tested by the questions. Because QSR also depends on the reader, we evaluate sensitivity across five frozen language models.

The protocol also accounts for embedded message capacity and sampling uncertainty. Sentence-level evaluation is restricted to methods that support the required embedded message length, and each method retains its supported message length. Where bootstrap confidence intervals are reported, the resampling procedure and quantities held fixed are specified in the evaluation details.

B METRIC DEFINITIONS AND STATISTICAL PROPERTIES

AURSC. Let $\mathcal{A}$ denote the set of all attack types, and let $\widetilde{R}_a(d)$ be the TPR@1%FPR curve for attack type a at mean ASI d. Each curve includes the unattacked result at ASI $= 0$, uses linear interpolation between measured points, and is held constant beyond the largest measured ASI. For a reporting budget $b > 0$, we define

$$\mathrm{AURSC}(b) = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \frac{1}{b} \int_0^b \widetilde{R}_a(d)\, \mathrm{d}d.$$

Since each curve takes values in $[0, 1]$, AURSC also lies in $[0, 1]$. If one method's curve is at least as high as another's for every attack type and every $d \in [0, b]$, its AURSC is also at least as high. Equivalently, AURSC is the expected value of the constructed curve when the attack type is sampled uniformly from $\mathcal{A}$ and the ASI value is sampled uniformly from $[0, b]$. This interpretation concerns the interpolated curves; it does not imply that detection performance was measured at every ASI value. We use $b = 0.2$ throughout.

Observed minimum. Let $r_{a,j}$ and $d_{a,j}$ denote the measured TPR@1%FPR and mean ASI for condition j of attack type a. The observed minimum within budget b is

$$\mathrm{ObsMin}(b) = \min_{\substack{a \in \mathcal{A},\, j \\ d_{a,j} \le b}} r_{a,j},$$

provided that at least one measured condition satisfies the budget. Unlike AURSC, this statistic uses only measured conditions and does not interpolate between them. An attack type with no measured condition inside the budget does not contribute to the observed minimum. Consequently, observed Min@0.2 does not bound performance at unmeasured conditions within $[0, 0.2]$.

Codeword matching under independent bit errors. Consider a codebook of distinct binary codewords of equal length L. Suppose each embedded bit is independently flipped with the same probability $0 < p < 1/2$. For a decoded embedded message y and candidate codeword c,

$$P(y \mid c) = p^{d_H(y,c)} (1 - p)^{L - d_H(y,c)},$$

where d_H is the Hamming distance. This likelihood decreases as $d_H(y, c)$ increases. Selecting the codeword with the highest fractional bit agreement is therefore maximum-likelihood decoding under this model. Our attribution procedure additionally rejects predictions whose matching score does not exceed the calibrated threshold. The maximum-likelihood result concerns codeword selection under the stated error model; it does not establish optimality of the open-set rejection rule or of decoding under structured errors.

Calibration FPR and out-of-sample control. Let $s_{(1)} \ge \cdots \ge s_{(n)}$ be the pooled negative scores sorted in decreasing order. For a target FPR α, set

$$k = \lfloor \alpha n \rfloor, \qquad \tau = s_{(k)},$$

with $k \geq 1$, and accept an image only if its score exceeds τ. At most $k-1$ calibration scores exceed this threshold, so the empirical pooled calibration FPR satisfies

$$\widehat{\mathrm{FPR}}_{\mathrm{cal}} \leq \frac{k-1}{n} \leq \alpha.$$

This calibration-set bound does not require independence or exchangeability.

If the n calibration scores and a fresh negative score S_{new} are jointly exchangeable, then

$$P(S_{\mathrm{new}} > \tau) \leq \frac{k}{n+1} \leq \alpha.$$

The first inequality is an equality when ties occur with probability zero. This is a marginal guarantee over the calibration sample and the fresh negative, not a guarantee conditional on a particular fitted threshold.

In our evaluation, multiple attacked-unwatermarked images originate from the same source image. Joint exchangeability of all pooled scores is therefore not assumed, and the out-of-sample bound is not claimed for this pool without additional justification. The empirical calibration bound still holds. Neither statement guarantees an FPR of at most α for every attack condition or negative subgroup.

Bit accuracy and complete-message recovery. Bit accuracy records the proportion of correctly decoded bits but does not identify their positions. embedded messages with the same bit accuracy can therefore contain different character errors and yield different answers to the stored questions.

Under independent bit errors with a common probability p, expected bit accuracy is $1-p$. For an ℓ-character field represented by 5ℓ bits, the probability of recovering every bit correctly is

$$P(\text{exact bit recovery}) = (1-p)^{5\ell}.$$

With a one-to-one character encoding and no additional error correction or text normalization, this is also the probability of exact field recovery. This calculation illustrates the distinction between bit accuracy and complete recovery; it does not assume that the evaluated watermark decoders produce independent bit errors.

Question Success Rate. Let $\hat{t}$ be the decoded text, q_i a stored question, a_i its gold answer, and ρ a frozen reader. Let $N(\cdot)$ denote answer normalization: lower-casing, replacing punctuation with spaces, and removing articles. Define $C(y, a)$ to equal one if $N(y)$ exactly matches $N(a)$, or if $N(y)$ contains all gold-answer tokens and has at most six tokens; otherwise, let $C(y, a) = 0$. This is the scoring rule described in Appendix D.2.

For a message with Q stored questions,

$$\mathrm{QSR}(\hat{t}) = \frac{1}{Q} \sum_{i=1}^{Q} C\big(\rho(\hat{t}, q_i), a_i\big)\,.$$

QSR measures the fraction of questions answered correctly from the recovered text under the specified reader and scoring rule. It depends on both the recovered embedded message and the reader, and does not measure recovery of information that the stored questions do not test. The measured empty-message QSR is zero for every evaluated reader and evaluation subset. No baseline subtraction is applied.

C Attack grid and chain specification

Table 5 lists the attack category 1 conditions and attack category 2 single attacks with their mean ASI on the matched synthetic cohort for both watermark families. Table 6 lists the fixed chains used in attack categories 3 and 4.

D Semantic corpus details

The corpus contains 240 messages, with 120 messages in each length class. Long messages contain at most 28 characters and four stored questions; short messages contain at most 11 characters and three stored questions. Question types cover entities, relations, attributes, and temporal information. Every gold answer is a single token and every message describes fictional people. Messages are encoded at 5 bits per character over a 32-symbol alphabet.

Table 5: Single-attack conditions. Mean ASI is averaged over images and methods on the matched synthetic cohort, separately for post-hoc and in-generation methods.

Attack	Parameter	Severities	Mean ASI, post-hoc (synthetic)	Mean ASI, in-generation (synthetic)
brightness	factor	0.7, 1.2, 1.5	0.052, 0.027, 0.110	0.060, 0.031, 0.124
contrast	factor	0.7, 1.2, 1.5	0.040, 0.018, 0.074	0.046, 0.021, 0.084
hue	shift (degrees)	10, 20, 30	0.023, 0.065, 0.110	0.024, 0.066, 0.111
overlay	opacity	0.1, 0.2, 0.3	0.159, 0.264, 0.349	0.143, 0.244, 0.324
Gaussian noise	σ (8-bit levels)	5, 10, 20	0.115, 0.268, 0.508	0.161, 0.353, 0.630
Gaussian blur	kernel size	5, 7, 9	0.200, 0.276, 0.321	0.165, 0.229, 0.267
resize	scale factor	0.7, 0.5, 0.3	0.084, 0.168, 0.272	0.071, 0.142, 0.229
JPEG	quality	70, 50, 30	0.039, 0.067, 0.117	0.047, 0.080, 0.140
crop	kept fraction of each side	0.8, 0.7, 0.5	0.458, 0.549, 0.680	0.524, 0.604, 0.716
rotation	angle (degrees)	5, 10, 15	0.267, 0.399, 0.477	0.328, 0.465, 0.535
VAE regeneration	quality factor	3, 1	0.185, 0.248	0.258, 0.329
diffusion regeneration	strength	0.1, 0.2, 0.3	0.149, 0.229, 0.303	0.112, 0.154, 0.200
prompt-based edit	editor	Qwen-Image-Edit, FLUX.2-klein	0.216, 0.237	0.265, 0.287
UnMarker	default	–	0.361	0.433

Table 6: The fixed attack chains of attack categories 3 and 4. Mean ASI is averaged over images and methods on the matched synthetic cohort, separately for post-hoc and in-generation methods.

Chain	Attack category	Stages	Mean ASI, post-hoc (synthetic)	Mean ASI, in-generation (synthetic)
`messenger`	C3	resize 0.7 → JPEG 70	0.123	0.115
`filter`	C3	brightness 1.2 → contrast 1.2	0.059	0.066
`social`	C3	crop 0.9 → resize 0.7 → JPEG 70	0.400	0.445
`reshare`	C3	resize 0.7 → JPEG 70 → resize 0.9 → JPEG 50	0.218	0.202
`straighten`	C3	rotation 5 → crop 0.9 → resize 0.7 → JPEG 70	0.457	0.497
`filterpost`	C3	brightness 1.2 → saturation 1.2 → sharpen 1.5 → JPEG 70	0.171	0.182
`noisedenoise`	C3	noise 5 → blur 5 → JPEG 70	0.218	0.192
`meme`	C3	overlay 0.1 → resize 0.7 → JPEG 70	0.249	0.235
`editshare-flux2`	C4	edit (FLUX.2-klein) → resize 0.7 → JPEG 70	0.299	0.336
`editshare-qwen`	C4	edit (Qwen-Image-Edit) → resize 0.7 → JPEG 70	0.285	0.317
`vaeregen`	C4	VAE regen. → JPEG 70	0.195	0.277
`regenshare`	C4	diffusion regen. → crop 0.9 → JPEG 70	0.450	0.441
`scrub`	C4	diffusion regen. → noise 2 → blur 3 → sharpen 1.0 → JPEG 70	0.332	0.275
`unmarkshare`	C4	UnMarker → resize 0.7 → JPEG 70	0.423	0.477

D.1 Message encoding

Each message is converted to bits with a fixed-width code of 5 bits per character over a 32-symbol alphabet: the 26 lowercase letters, the space, and the five punctuation marks `. , ' - ?`. The corpus is written within this alphabet, so no character is lost. Before encoding, every message is padded with spaces to the character budget of its length class, 11 characters for short and 28 characters for long messages, so that all encoded messages of a class have the same length: 55 bits for short and 140 bits for long messages. The encoded bits are then zero-padded to the capacity of the watermarking method, so that the payload has the method's native length. Each method carries the largest class whose encoded length fits its capacity, which assigns the long class to methods with 256-bit capacity and the short class to methods with 64- or 100-bit capacity (Table 4).

Decoding reverses the process on the recovered bits: each group of 5 bits is mapped back to a symbol, trailing padding spaces are removed, and the zero padding beyond the character field is ignored. Any recovered bit string therefore yields a string over the alphabet, so a corrupted payload produces corrupted text rather than a decoding failure, and the reader sees the corruption. Because the code has fixed width, a bit error changes only the character that contains it and does not shift the remaining characters. Bit accuracy is computed over the full payload, including the zero padding. A 5-bit code over a 32-symbol alphabet is the same arrangement as the Base32 encoding (Josefsson, 2006).

D.2 How an answer is scored

The reader is instructed to answer from the recovered text only and to return `unknown` when the text does not determine the answer. Scoring is deterministic: answers are lower-cased, punctuation

is replaced by spaces, and articles are removed. An answer is correct if it equals the gold answer exactly or contains every gold token within at most six tokens. Table 7 gives examples.

Table 7: Worked scoring examples for the message "wanda moved the green ladder".

Gold answer	Reader answer	em	correct	Reason
wanda	"Wanda."	✓	✓	case and punctuation stripped
wanda	"the Wanda"	✓	✓	article dropped
wanda	"it was wanda"	×	✓	contains gold token, $\leq$ 6 tokens
wanda	"wanna"	×	×	no fuzzy matching
wanda	"unknown"	×	×	reader failure
ladder	"a green ladder"	×	✓	extra words allowed
ladder	"ladders"	×	×	different token
ladder	"green"	×	×	answers a different question

E PROTOCOL DETAILS

E.1 EVALUATION DATA AND IMPLEMENTATION DETAILS

Generation settings. SD2.1 renders at 512 px and the other generators at 1024 px. SD2.1 and SDXL use 50 sampling steps with guidance 7.5, except PRC, which uses guidance 3.0 as in its reference implementation. FLUX.1-dev uses 50 steps with guidance 3.5. Complete generator configurations are included in the released metadata.

ASI under geometric attacks. A cropped image is scaled back to the reference size before LPIPS comparison, and a rotated image retains its transformed canvas. No inverse warp or registration is applied. ASI therefore includes the geometric displacement present in the image received by the watermark decoder.

Evaluation subsets. Evaluation subsets are content-stratified, with 200 images per subclass and category. The synthetic subset contains 1,593 images, of which 1,193 generated by SD2.1, SDXL, and FLUX.1 form the matched synthetic cohort. UnMarker is evaluated on a nested subset of 50 images per subclass.

Evaluation subsets and calibration. For cross-family comparisons, the 1,193 images of the matched synthetic cohort are evaluated together as a separate subset. Thresholds are recalibrated using this cohort's own unwatermarked and attacked-unwatermarked negatives.

The pooled calibration set of the matched cohort contains 61,077 negatives per method (60,843 on the sentence-payload images), and the realized pooled calibration FPR is 0.28-0.93% (0.48-0.91%). The in-generation evaluation uses the same 1,193 images. Differences between the matched and the pooled synthetic results reflect both the change of population and the recalibration.

E.2 DETECTION CALIBRATION

Detection scores. Each method is evaluated using its detector score, with the score direction standardized so that larger values indicate stronger evidence for a watermark. For post-hoc methods on the identity-payload subsets, the score is the highest fractional bit agreement between the decoded message and the 100 identity codewords (Appendix E.3). In the family comparison of Table 3, the seven sentence-capable post-hoc methods are evaluated on the sentence-payload images and scored by the fractional bit agreement with the embedded payload, so that both families are tested against a known payload. Threshold calibration is performed separately for each method and evaluation subset.

Negative pools. For each method and evaluation subset, the calibration pool contains unattacked unwatermarked images and attacked-unwatermarked images. An attacked-unwatermarked image is obtained by applying an attack condition to the corresponding unwatermarked image. Attacked-unwatermarked images from all evaluated attack conditions in a subset are included in the same pool. The resulting threshold is therefore shared across attack conditions rather than recalibrated for each attack.

Threshold calculation. Let

$$s_{(1)} \geq s_{(2)} \geq \cdots \geq s_{(n)}$$

denote the n pooled negative scores sorted in decreasing order. For target false-positive rate $\alpha = 0.01$, we set

$$k = \lfloor \alpha n \rfloor, \qquad \tau = s_{(k)},$$

and accept an image only when its score is strictly greater than τ. With this rule, at most $k - 1$ calibration negatives are accepted and the empirical FPR of the pooled calibration set is therefore at most 1%. All calibration pools contain enough negative examples for $k \geq 1$.

The same threshold is used for the unattacked condition and all attacked conditions within that method and evaluation subset. Because calibration is performed on the pooled negative distribution, the 1% target applies to the pool as a whole and does not require every attack condition or negative subgroup to have an individual FPR of 1%.

unwatermarked-only calibration diagnostic. We additionally calibrate a separate threshold using unattacked negatives only and evaluate this threshold on attacked-unwatermarked images. This diagnostic measures changes in false-positive rate caused by image transformations when transformed negatives are excluded from calibration. These unwatermarked-only thresholds are used only for this diagnostic and are not used for the headline TPR@1%FPR results. Table 25 reports the corresponding false-positive rates.

E.3 Identity codebooks and attribution sensitivity

Identity codebooks. We use $N = 100$ candidate identities as a common attribution setting and do not interpret this value as the maximum identity population supported by a watermarking method. For nine post-hoc methods, identity codewords are uniformly random bit strings generated using fixed seeds. TrustMark and VINE instead encode random alphanumeric strings using their native embedded message encoders, leaving 6 of 48 and 16 of 100 bit positions fixed, respectively.

The codebooks are constructed without a minimum-distance constraint. Among the 100 identities, the minimum pairwise Hamming distance is 5-6 bits for the 30-48-bit methods, 16 bits for EditGuard, 21-32 bits for the 100-bit methods, and 95-97 bits for the 256-bit methods. These distances describe codeword separation only; attribution performance also depends on the errors produced by each decoder.

Open-set attribution. For each image, the recovered embedded message is compared with every candidate codeword using fractional bit agreement, defined as the fraction of matching bit positions. The image is assigned to the codeword with the highest agreement. The assignment is accepted only when this score exceeds the threshold calibrated for that method and evaluation subset. A watermarked image is counted as correctly attributed only when it is accepted and assigned to its embedded identity. Rejected images and accepted assignments to a different identity are both counted as attribution errors.

Attribution errors at $N = 100$. Attribution errors can therefore be separated into rejections and accepted wrong-identity assignments. We aggregate attacked conditions and synthetic subsets by image count within each method and then average equally across methods. The mean rejection rates are 29.7%, 30.6%, and 29.1% on manipulated, real, and synthetic images, respectively. The corresponding accepted wrong-identity rates are 0.3%, 0.2%, and 0.3%. The largest method-level mean wrong-identity rate across these categories is 1.4%, for WAM on manipulated images. These quantities are averages over attacked conditions and are not bounds for individual attack conditions.

Evaluation with $N = 10$. As a sensitivity analysis, we additionally evaluate the first 10 identities of each codebook. Only images carrying these identities are treated as positives. Images carrying the remaining 90 identities become out-of-set negatives, and the attribution threshold is recalibrated for the $N = 10$ evaluation.

Mean open-set attribution accuracy across methods is 0.711 versus 0.700 on manipulated images, 0.701 versus 0.692 on real images, and 0.720 versus 0.707 on synthetic images for $N = 10$ and $N = 100$, respectively. The smaller codebook is therefore 0.9-1.3 percentage points more accurate on average. The difference is 0.6 percentage points under attack category 1 and 0.9-2.6 percentage points under attack categories 2-4 on all three image categories. The ordering of the 11 methods

is identical at both codebook sizes on manipulated images and differs by one exchange of adjacent methods on real and synthetic images, in both cases between methods separated by at most 1.0 percentage points. The largest method-level difference is 5.1 percentage points, for PIMoG on synthetic images. PIMoG has 30-bit codewords with the smallest pairwise separation among the codebooks, and its differences are 4.3-5.1 percentage points on the three image categories.

The composition of the attribution errors is the same at both codebook sizes. Every method attributes unattacked images with accuracy 0.999 or higher at both $N = 10$ and $N = 100$. On attacked images, the rejection rate is 28-31% at both sizes, while the accepted wrong-identity rate is 0.1% at $N = 10$ and 0.2-0.3% at $N = 100$. Increasing the candidate set tenfold therefore adds fewer than 0.2 percentage points of identity confusion, and attribution errors are rejections caused by the attack rather than confusions between identities.

The $N = 10$ evaluation changes the candidate identity set, the positive subset, and the calibration negatives, so the differences above bound the combined effect of these changes rather than isolating codebook size, and neither evaluation characterizes deployment-scale identity populations. For the comparisons made in this paper, however, $N = 100$ is sufficient. The effect of codebook size on attribution accuracy is more than an order of magnitude smaller than the differences between methods and between attack categories, it leaves the method ordering and the identification of damaging attacks unchanged, and the reported accuracies reflect watermark survival under attack rather than codebook capacity.

False positives across negative subgroups. The pooled 1% calibration target does not imply the same false-positive rate for every negative subgroup. For example, on manipulated images in the $N = 10$ evaluation, VINE has FPRs of 0.00%, 0.00%, and 1.85% on unattacked, attacked-unwatermarked, and out-of-set negatives, respectively. VideoSeal has corresponding rates of 1.52%, 1.42%, and 0.45%. These subgroup rates are reported as diagnostics and are not separately calibrated operating points.

E.4 Robustness metrics and reporting budget

Curve construction. For each method, evaluation subset, and attack type, every evaluated condition gives one point with mean ASI on the x-axis and TPR@1%FPR on the y-axis. Let $d_{a,j}$ denote the mean ASI and $r_{a,j}$ the TPR@1%FPR for condition j of attack type a. We also include the unattacked result at ASI $= 0$.

For each attack type, we order the measured points by mean ASI and connect adjacent points using linear interpolation. This gives a piecewise-linear curve $\widetilde{R}_a(d)$. If the largest measured ASI is below the reporting budget, the curve is held constant from the last measured point to the budget boundary. The same calibrated detection threshold is used for all conditions within an evaluation subset.

AURSC. We compute AURSC over all the attack types: the attacks of attack categories 1 and 2 in Table 5 and the processing chains of attack categories 3 and 4 in Table 6.

Let $\mathcal{A}$ denote this set of attack types. For an ASI reporting budget $b > 0$,

$$\mathrm{AURSC}(b) = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \frac{1}{b} \int_0^b \widetilde{R}_a(d)\, \mathrm{d}d.$$

The factor $1/b$ normalizes the area so that AURSC lies in $[0, 1]$. We first compute one normalized area for each attack type and then average equally across all attack types. An attack type therefore receives the same weight regardless of how many parameter settings are evaluated.

All attack types contribute to AURSC, including attack types for which no measured condition has mean ASI within the reporting budget.

Observed Min@0.2. For a reporting budget b, we also report the lowest TPR@1%FPR that is directly observed at a measured condition with mean ASI at most b:

$$\mathrm{ObsMin}(b) = \min_{\substack{a \in \mathcal{A},\, j \\ d_{a,j} \le b}} r_{a,j}.$$

Table 8: Sensitivity of AURSC method rankings to the ASI reporting budget. Post-hoc columns report Spearman's ρ relative to the ranking at $b = 0.2$. The final column reports whether the in-generation ranking changes relative to $b = 0.2$.

Budget b	Post-hoc, manipulated	Post-hoc, real	In-generation ranking
0.10	1.000	1.000	Unchanged
0.30	0.991	0.991	Unchanged
0.50	0.982	0.982	Unchanged

The main results use $b = 0.2$, and we refer to this value as *Observed Min@0.2*. Only measured attack conditions with mean ASI ≤ 0.2 are included. If an attack type has no measured condition inside this range, it does not contribute to Observed Min@0.2.

AURSC and Observed Min@0.2 therefore summarize different properties. AURSC summarizes detection performance across the full ASI range, using interpolation where required. Observed Min@0.2 reports the lowest TPR@1%FPR that is directly measured within the same range.

Measured coverage of the ASI range. We assess coverage on the matched synthetic cohort for both post-hoc and in-generation methods. Not all attacks have a measured condition with mean ASI ≤ 0.2, and the same attack can fall inside the reporting range for one family and outside it for the other.

For the processing chains in Table 6, four chains have family-average ASI ≤ 0.2 in each family. Messenger, filter, and filterpost fall within the range for both families. Vaeregen falls within the range for post-hoc methods (ASI 0.195) but outside it for in-generation methods (0.277). Conversely, noisedenoise falls outside the range for post-hoc methods (0.218) but within it for in-generation methods (0.192). Reshare lies outside the range for both families, with ASI 0.218 and 0.202, respectively.

These comparisons use family-average ASI and need not hold for every method. All attacks contribute to AURSC, while each method's curves and observed Min@0.2 use its own condition-level mean ASI.

When an attack's lowest measured mean ASI exceeds 0.2, its curve within the reporting range is obtained by linear interpolation between the unattacked result at ASI $= 0$ and the measured point with the lowest mean ASI. The attack contributes to AURSC but not to the observed minimum.

When an attack's highest measured mean ASI is below 0.2, its curve is held constant from that point to the reporting boundary. Each fixed processing chain has one measured attacked condition, so its curve uses that condition and the unattacked result.

Reporting Robustness. We use $b = 0.2$ as a fixed ASI range for reporting robustness. We do not interpret 0.2 as a universal threshold for acceptable image change. Rather, it provides one common range over which all methods are summarized. The measured attack grid provides partial coverage of this range. Table 6 reports the mean ASI of processing chains on the matched synthetic cohort for both families. These values describe the scale of the reporting budget; they do not establish a threshold for acceptable image change.

For geometric attacks, ASI includes the spatial displacement introduced by the transformation because images are not realigned before computing LPIPS. The family-average ASI values of all evaluated cropping and rotation settings are above 0.2. Observed Min@0.2 nevertheless uses each method's own condition-level mean ASI when deciding whether a measured condition is inside the reporting range. The same rule is used for method-dependent attacks such as UnMarker.

Table 9: Post-hoc robustness across the three image categories. Unattacked is TPR@1%FPR (%) before attack. AURSC is the normalized area under the TPR@1%FPR-versus-ASI curve over $[0, 0.2]$, and Obs. Min@0.2 is the minimum observed TPR@1%FPR within ASI ≤ 0.2, both over all attacks. Attr. acc. is mean open-set attribution accuracy at 1% FPR over all attacked conditions. Max FPR is the highest realized FPR on attacked-unwatermarked negatives when the threshold is calibrated using unattacked negatives only.

		manipulated					Real					Synthetic				
Method	Bits	Unatt.	AURSC	Obs. Min@0.2	Attr. acc.	Max FPR	Unatt.	AURSC	Obs. Min@0.2	Attr. acc.	Max FPR	Unatt.	AURSC	Obs. Min@0.2	Attr. acc.	Max FPR
CIN	30	100.0	0.696	0.002	46.0	0.6	100.0	0.699	0.003	46.3	0.8	100.0	0.682	0.000	45.9	0.9
PIMoG	30	100.0	0.801	0.016	59.4	0.6	100.0	0.807	0.015	60.2	0.6	100.0	0.789	0.004	58.7	1.1
WAM	32	100.0	**0.975**	**0.965**	**90.2**	2.0	100.0	0.960	0.766	86.3	2.8	100.0	**0.968**	0.771	**89.1**	5.6
TrustMark	48	99.9	0.868	0.006	66.4	1.4	100.0	0.878	0.009	66.0	1.4	99.9	0.863	0.005	64.2	1.4
EditGuard	64	100.0	0.721	0.002	40.3	0.8	99.9	0.746	0.004	40.9	1.9	100.0	0.714	0.000	41.3	0.9
VINE	100	100.0	0.929	0.936	77.7	1.8	100.0	0.941	**0.933**	77.6	0.7	100.0	0.942	**0.971**	80.3	1.7
InvisMark	100	100.0	0.823	0.011	66.6	3.4	100.0	0.797	0.003	59.7	1.2	100.0	0.890	0.000	74.1	5.0
OmniGuard	100	100.0	0.943	0.572	81.2	0.7	100.0	0.955	0.445	81.3	1.1	100.0	0.937	0.157	81.3	1.9
MBRS	256	100.0	0.880	0.005	69.7	1.7	100.0	0.901	0.009	70.9	1.5	100.0	0.864	0.001	69.3	1.4
VideoSeal	256	100.0	0.954	0.015	86.8	1.1	100.0	**0.962**	0.057	**86.6**	1.1	100.0	0.953	0.013	86.2	1.7
PixelSeal	256	100.0	0.942	0.007	85.5	1.6	100.0	0.949	0.025	85.3	1.2	100.0	0.945	0.010	86.7	1.5

F Additional results

F.1 Post-hoc robustness across image categories

Table 9 gives the complete post-hoc results for manipulated, real, and synthetic images.

Across the 11 post-hoc methods and three image categories, AURSC ranges from 0.682 to 0.975. Within ASI ≤ 0.2, the same 8 methods fall below 7% TPR@1%FPR under at least one attack in every category, and 5-6 fall below 1%. Only WAM and VINE remain above 50% at every measured condition with mean ASI ≤ 0.2 in all three image categories.

Mean open-set attribution accuracy over attacked conditions remains within 1.4 percentage points of mean TPR@1%FPR for every method and category. Per-method AURSC differs by at most 0.027 between manipulated and real images and by 0.067 between manipulated and synthetic images.

F.2 Detection in identity and sentence mode

Table 10 compares the two modes for the seven post-hoc methods that carry sentences. Sentence mode raises the seven-method mean TPR@1%FPR over all attacked conditions by 0.6-1.0 percentage points and the category-2 mean by 4.2-4.9 points, and it does not change any ordering or conclusion reported in the paper. InvisMark is the only method that is lower in sentence mode: its decoder returns zeros on unmarked images, and the zero padding of the sentence payload raises its unmarked scores and therefore its threshold.

F.3 Attack-category and per-condition robustness

The four attack categories show similar aggregate trends across watermark families. Mean post-hoc TPR@1%FPR is 81-83% at attack category 1, 42-45% at attack category 2, 69-73% at attack category 3, and 39-41% at attack category 4. The corresponding in-generation values are 79.5%, 71.0%, 73.4%, and 52.2%.

For sentence embedded messages, the same ordering appears in bit and message recovery. On synthetic images, post-hoc QSR decreases from 0.565 at attack category 1 to 0.153 at attack category 2, complete-message recovery decreases from 0.508 to 0.111, and bit accuracy decreases from 0.884 to 0.703. For in-generation methods, the corresponding changes are 0.737 to 0.354 for QSR, 0.737 to 0.236 for complete-message recovery, and 0.908 to 0.838 for bit accuracy.

Model-based attacks produce several of the largest failures, but individual methods respond differently. At the stronger VAE regeneration setting, 6-7 of the 11 post-hoc methods fall below 10%

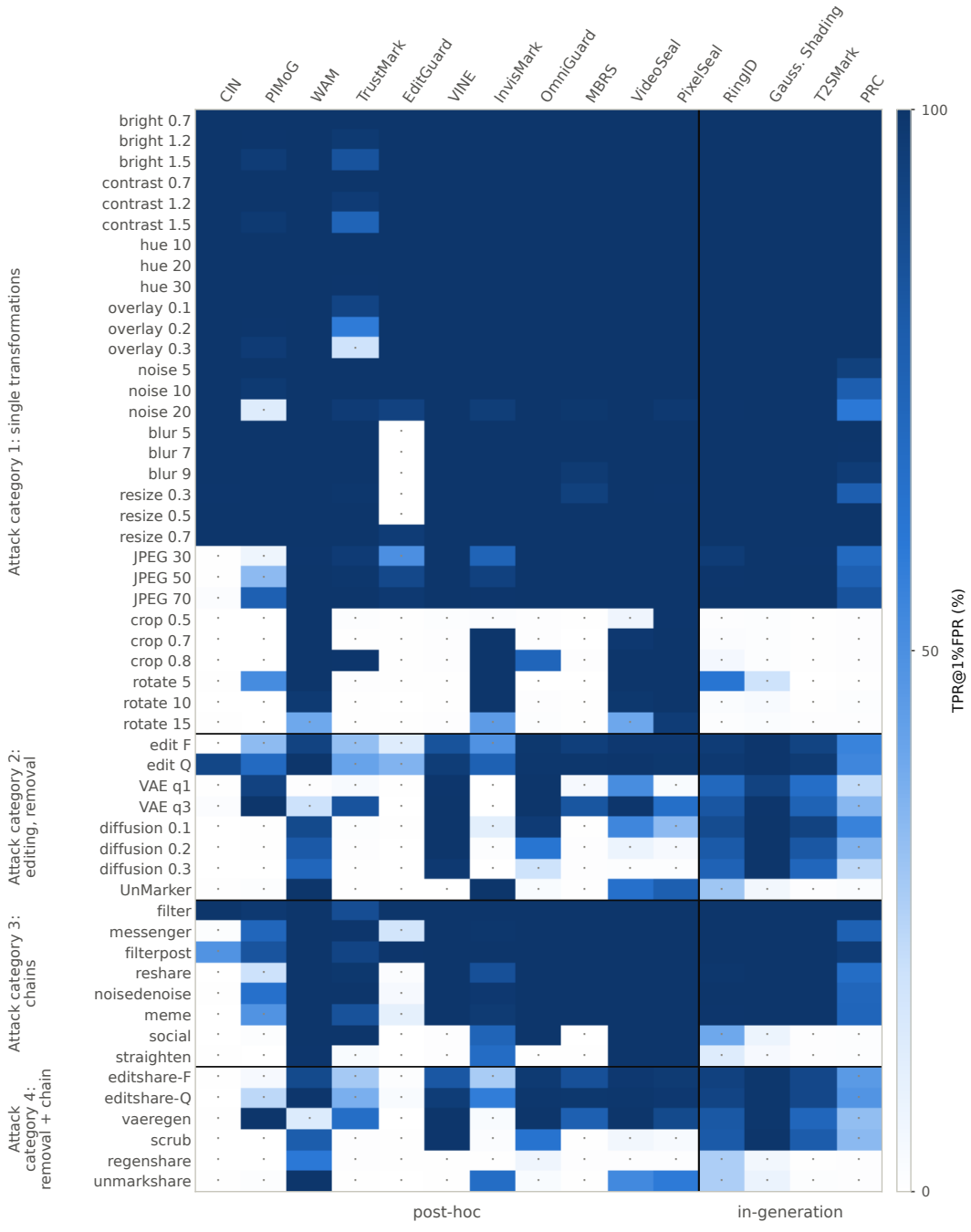

Figure 3: TPR@1%FPR for each attack condition on the synthetic category for the 11 post-hoc methods in identity mode (left) and 4 in-generation methods (right); the matched-payload family comparison is Table 3. Each row is one of the 52 attack conditions, grouped by attack category, and a dot marks values below 50%.

Table 10: The seven message-capable post-hoc methods in identity mode and in message mode. Identity mode is the open-set codebook score on the identity-payload images, the basis of the identity-arm block of Table 3 and of the other detection tables in this appendix, the basis of the semantic-arm block of Table 3. For each image category, All is the mean TPR@1%FPR (%) over all attacked conditions, C2 is the mean over the category-2 conditions, and Obs. Min@0.2 is the lowest measured TPR@1%FPR within the ASI budget.

		Synthetic (matched)			Real			Manipulated		
Method	Mode	All	C2	Obs. Min@0.2	All	C2	Obs. Min@0.2	All	C2	Obs. Min@0.2
EditGuard	identity	42.4	7.6	0.000	41.1	5.8	0.004	40.5	6.5	0.002
	message	47.7	12.7	0.000	45.7	10.8	0.009	45.9	12.1	0.005
VINE	identity	81.0	95.4	0.972	77.9	83.1	0.933	77.9	84.9	0.936
	message	81.4	97.4	0.999	78.0	83.8	0.955	78.2	85.3	0.968
InvisMark	identity	74.2	19.6	0.001	60.1	12.1	0.003	67.2	19.8	0.011
	message	67.7	11.8	0.004	54.2	11.0	0.009	58.3	14.1	0.006
OmniGuard	identity	81.3	78.5	0.511	81.4	78.3	0.445	81.3	78.7	0.572
	message	84.0	88.5	0.816	83.5	85.6	0.792	83.7	86.7	0.823
MBRS	identity	70.5	39.8	0.003	71.3	42.6	0.009	69.9	39.4	0.005
	message	71.3	45.7	0.000	71.1	45.9	0.001	70.6	44.4	0.003
VideoSeal	identity	86.4	55.7	0.009	86.7	58.4	0.057	86.9	61.8	0.015
	message	89.7	68.2	0.067	89.4	68.7	0.231	91.0	75.9	0.120
PixelSeal	identity	87.5	43.2	0.015	85.4	39.4	0.025	85.6	42.7	0.007
	message	88.7	48.9	0.028	86.5	43.8	0.059	87.8	50.0	0.049
Mean	identity	74.8	48.5	0.216	72.0	45.7	0.211	72.8	47.7	0.221
	message	75.8	53.3	0.273	72.6	49.9	0.294	73.6	52.6	0.282

Table 11: Performance across the four attack categories: single perturbations (C1), editing and removal attacks (C2), processing chains (C3), and removal followed by a chain (C4). TPR@1%FPR is reported for both watermark families, post-hoc methods in identity mode and in-generation methods on their sentence-carrying images; the matched-payload family comparison is Table 3. Open-set attribution accuracy is additionally reported for post-hoc methods. QSR, complete-message recovery (Msg. Rec.), and bit accuracy (Bit Acc.) are computed for methods that carry sentence embedded messages. Unattacked is the result before attack, and All is the mean over all attacked conditions.

	Unattacked	C1	C2	C3	C4	All
TPR@1%FPR (%)						
post-hoc, fake	100.0	81.9	44.7	71.1	40.4	70.3
post-hoc, real	100.0	81.3	42.1	68.7	38.8	69.4
post-hoc, synthetic	100.0	82.6	43.8	73.4	40.7	70.9
in-generation, synthetic	100.0	79.5	71.0	73.4	52.2	75.3
Open-set attribution accuracy at 1% FPR (%)						
post-hoc, fake	100.0	81.7	44.2	70.9	39.6	70.0
post-hoc, real	100.0	81.1	41.7	68.5	38.3	69.2
post-hoc, synthetic	100.0	82.5	43.3	73.3	40.0	70.7
QSR						
post-hoc, fake	0.821	0.552	0.156	0.492	0.146	0.442
post-hoc, real	0.816	0.536	0.146	0.457	0.130	0.425
post-hoc, synthetic	0.825	0.565	0.153	0.504	0.141	0.450
in-generation, synthetic	0.984	0.737	0.354	0.647	0.236	0.635
Msg. Rec.						
post-hoc, fake	0.845	0.498	0.114	0.412	0.105	0.387
post-hoc, real	0.836	0.475	0.106	0.357	0.097	0.364
post-hoc, synthetic	0.852	0.508	0.111	0.415	0.106	0.392
in-generation, synthetic	1.000	0.737	0.236	0.627	0.148	0.610
Bit Acc.						
post-hoc, fake	0.990	0.878	0.700	0.846	0.683	0.826
post-hoc, real	0.989	0.873	0.691	0.836	0.676	0.821
post-hoc, synthetic	0.991	0.884	0.703	0.863	0.689	0.833
in-generation, synthetic	1.000	0.908	0.838	0.879	0.762	0.883

TPR@1%FPR in each category, while the in-generation mean remains at 63%. Gaussian Shading retains 93%, whereas PRC falls to 22%.

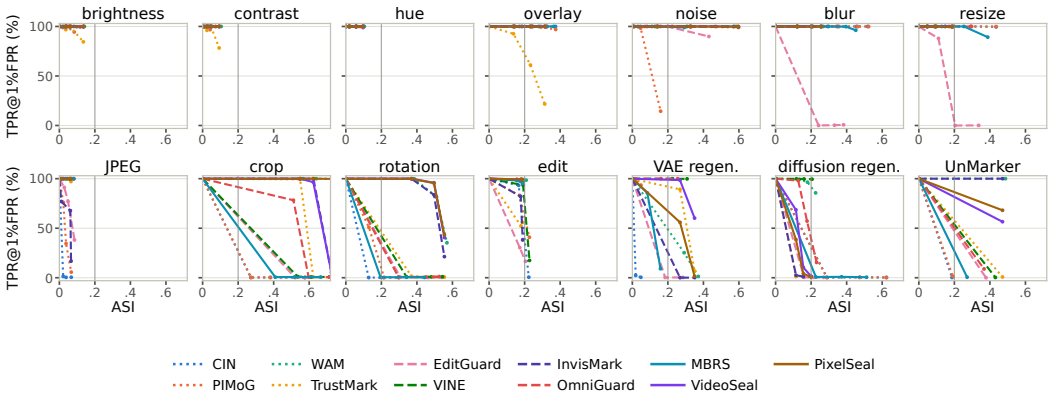

Figure 4: TPR@1%FPR against visible damage (ASI) on the manipulated category for the 11 post-hoc methods in identity mode. AURSC is measured over ASI $\in [0, 0.2]$.

At diffusion regeneration strength 0.3, only WAM and VINE among the post-hoc methods remain above 20% TPR@1%FPR in every category, giving a post-hoc mean of 12-19%. Gaussian Shading, RingID, and T2SMark retain 73-100%, with an in-generation mean of 68%. UnMarker gives a different ordering: 7 of the 11 post-hoc methods fall below 10%, while WAM and InvisMark remain at 100%, and Gaussian Shading, T2SMark, and PRC fall to 1-5% among the in-generation methods.

Additional equal-ASI comparisons show the same separation between visual damage and watermark removal. Gaussian noise with $\sigma = 20$ reaches mean ASI 0.459 on manipulated images while retaining 91.1% TPR@1%FPR, whereas diffusion regeneration at strength 0.3 has lower mean ASI, 0.317, but reduces TPR@1%FPR to 18.7%. For in-generation methods, Gaussian noise with $\sigma = 20$ reaches mean ASI 0.630 while retaining 89.6% TPR@1%FPR.

For attack category 3 chains, composition causes little additional loss after measured damage is matched. No post-hoc chain falls more than 0.2 percentage points below its hardest component, while six of the eight in-generation chains are within 4 points of their hardest component. The two quantities are evaluated within their respective watermark families and are not used for cross-family comparison.

F.4 Generator and content robustness

Gaussian Shading has the highest Min@0.2 (0.999) and AURSC (0.947) among the four in-generation methods and remains stable across generators. PRC falls from 76.8% mean TPR@1%FPR on SD2.1 to 47.6% on FLUX.1. RingID and T2SMark also show lower robustness on FLUX.1 than on SD2.1 or SDXL.

RingID additionally supports identity attribution. Across all attacks, it reaches 80.1% TPR@1%FPR for detection and 54.1% attribution accuracy over its native 1,024-key codebook.

Across the eight content subclasses, the median within-method range in post-hoc TPR@1%FPR is 1.7-2.1 percentage points per image category. The largest post-hoc range is 6.7 points for TrustMark on real images, while the largest in-generation range is 6.7 points for PRC.

Unattacked message recovery. Before attack, most methods recover the embedded message. Six of the seven post-hoc methods have unattacked QSR between 0.86 and 0.98, while the three in-generation methods have QSR between 0.983 and 0.984 (Table 4). InvisMark is the main exception: its unattacked bit accuracy is 0.937 but its complete-message recovery is 0.051 and its QSR is 0.240 because errors at fixed embedded message positions corrupt specific characters.

Complete-message recovery. Table 19 reports complete-message recovery next to bit accuracy and QSR. Complete-message recovery is the fraction of decoded sentences that exactly match the embedded sentence and does not use a reader. Under attack, it is 0.364-0.396 for post-hoc methods across image categories and cohorts and 0.610 for in-generation methods.

Table 12: AURSC on TPR@1%FPR for each attack type, averaged over post-hoc methods within each image category and over in-generation methods on synthetic images. Post-hoc methods are in identity mode; the matched-payload family comparison is Table 3.

Attack	manipulated	Real	Synthetic	In-generation
brightness	0.988	0.989	0.990	1.000
contrast	0.984	0.985	0.982	1.000
hue	0.999	0.999	0.999	1.000
overlay	0.993	0.991	0.992	1.000
Gaussian noise	0.961	0.961	0.971	0.990
Gaussian blur	0.962	0.967	0.953	1.000
resize	0.975	0.985	0.964	0.998
JPEG	0.740	0.734	0.799	0.942
crop	0.873	0.874	0.863	0.813
rotation	0.745	0.742	0.724	0.758
prompt-based edit	0.902	0.886	0.906	0.954
VAE regeneration	0.746	0.763	0.730	0.900
diffusion regeneration	0.614	0.617	0.567	0.853
UnMarker	0.770	0.687	0.754	0.791
`messenger`	0.863	0.854	0.876	0.959
`filter`	0.989	0.990	0.989	1.000
`social`	0.871	0.870	0.865	0.804
`reshare`	0.868	0.884	0.892	0.962
`straighten`	0.847	0.849	0.846	0.809
`filterpost`	0.949	0.934	0.955	0.995
`noisedenoise`	0.913	0.913	0.918	0.967
`meme`	0.898	0.895	0.912	0.972
`editshare-flux2`	0.828	0.842	0.851	0.949
`editshare-qwen`	0.848	0.863	0.874	0.946
`vaeregen`	0.777	0.791	0.772	0.902
`regenshare`	0.798	0.796	0.792	0.793
`scrub`	0.745	0.770	0.753	0.908
`unmarkshare`	0.819	0.805	0.810	0.810

Partial recovery. Table 20 separates QSR by whether the decoded sentence exactly matches the embedded sentence. For post-hoc methods on the matched synthetic cohort, 27.3% of correct answers come from messages that were not recovered exactly, compared with 5.1% for in-generation methods. Under the four additional readers, these shares are 26.1-29.5% and 4.8-5.6%. QSR on exactly recovered messages is below 1 because the reader can also make errors (Appendix F.5).

Likelihood-based check. ΔNLL is the reduction in the primary reader's negative log-likelihood of the embedded message when the recovered text is provided as context. It does not require generated answers, although it remains model-dependent. ΔNLL and QSR have Pearson's $r = 0.96$ across 2,385 method-attack pairs. This complementary likelihood-based check agrees with the QSR results.

F.5 Reader sensitivity

We repeat the complete reader pass of Section 4 with four additional instruction-tuned language models: Qwen3.5-9B (Qwen Team, 2026), Ministral-8B-Instruct-2410 (Mistral AI, 2024), Gemma-3-12B-IT (Gemma Team, 2025), and Llama-3.1-8B-Instruct (Grattafiori et al., 2024). Every reader receives the same recovered texts, stored questions, and prompt as the primary reader, generates at most 16 tokens with greedy decoding, and is scored with the rule of Appendix D.2, so that only the reader changes. All readers have frozen weights and are not fine-tuned. Qwen3.5-9B is the only reader with a thinking mode, and we disable it so that the answer is generated directly from the recovered text. Each reader scores the same 7.24 million question instances over all evaluation subsets. With an empty message, no reader answers any question correctly on any subset.

Table 13: Chain comparison at matched damage. Positive values indicate that the chain is harder than its strongest component at the same measured damage. Post-hoc columns use open-set attribution accuracy at 1% FPR; the in-generation column uses TPR@1%FPR. Mean ASI is measured on the manipulated category.

			Post-hoc (attribution acc.)			In-gen. (TPR)
Chain	Attack category	Mean ASI	Manipulated	Real	Synthetic	Synthetic
messenger	C3	0.099	-0.056	-0.071	-0.045	-0.004
filter	C3	0.065	+0.000	+0.002	-0.000	-0.000
social	C3	0.415	-0.024	-0.030	-0.001	+0.136
reshare	C3	0.191	-0.025	-0.025	-0.020	+0.007
straighten	C3	0.471	-0.092	-0.121	-0.068	+0.002
filterpost	C3	0.168	-0.245	-0.224	-0.191	-0.089
noisedenoise	C3	0.198	-0.096	-0.081	-0.068	-0.006
meme	C3	0.222	-0.101	-0.100	-0.092	-0.012
editshare (FLUX.2)	C4	0.251	+0.118	+0.113	+0.155	+0.116
editshare (Qwen)	C4	0.241	+0.043	+0.034	+0.064	+0.108
vaeregen	C4	0.183	-0.023	-0.030	-0.031	+0.031
regenshare	C4	0.465	+0.050	+0.033	+0.042	+0.144
scrub	C4	0.320	-0.039	-0.040	-0.050	-0.049
unmarkshare	C4	0.440	-0.014	+0.251	+0.042	+0.863

Table 14: In-generation robustness using TPR@1%FPR. AURSC is measured over $[0, 0.2]$, and Obs. Min@0.2 is the minimum observed TPR@1%FPR within the same damage budget. The final three columns report mean TPR@1%FPR over all attacked conditions for each generator.

			TPR@1%FPR by generator (%)		
Method	Obs. Min@0.2	AURSC	SD2.1	SDXL	FLUX.1
RingID	0.753	0.945	84.1	86.3	71.3
Gauss. Shading	0.999	0.947	82.2	80.3	79.6
T2SMark	0.726	0.927	79.9	80.1	69.9
PRC	0.360	0.862	76.8	66.6	47.6

Table 21 reports mean attacked QSR under each reader. The reader changes the absolute value of QSR: across the five readers, mean attacked QSR differs by at most 0.080. Under every reader, attacked QSR on the matched cohort is 0.14-0.20 higher for in-generation methods than for post-hoc methods.

Table 22 compares each additional reader with the primary reader. Condition-level QSR has Pearson's $r \geq 0.99$, and the readers agree with the primary reader on 93.3-94.2% of individual questions. Over all ten pairs of readers, r is at least 0.985, per-question agreement between 91.7% and 96.8%, and κ between 0.83 and 0.94.

Readers differ mainly in the absolute value of QSR (Table 23). Qwen3.5-9B and Gemma-3-12B give higher values than the primary reader, and Ministral-8B and Llama-3.1-8B give lower values. The ordering of the attack categories is the same under every reader and in every family and image category: QSR is highest at attack category 1, followed by attack categories 3, 2, and 4.

The order of the methods is partly reader-dependent (Table 24). In all three image categories and under all five readers, MBRS, EditGuard, and InvisMark have the three lowest attacked QSR values among post-hoc methods, in this order, and the in-generation order is T2SMark, Gaussian Shading, PRC. The remaining four post-hoc methods, PixelSeal, VideoSeal, VINE, and OmniGuard, always have the four highest values, but their order changes with the reader. VINE and OmniGuard carry short messages, on which the readers differ most.

To separate the reader from the watermark, we examine the questions whose post-hoc message was recovered exactly. For 11-character messages, the readers answer between 1.0% (Qwen3.5-9B) and 21.9% (Llama-3.1-8B) of these questions incorrectly, compared with 0.8-3.5% for 28-character

Table 15: In-generation methods by generator: TPR@1%FPR (%) for each attack category. Unattacked is the result before attack, C1-C4 are the four attack categories, and All is the mean over all attacked conditions.

Generator	Method	Unattacked	C1	C2	C3	C4	All
SD2.1	RingID	100.0	84.0	96.7	80.5	73.3	84.1
SD2.1	Gauss. Shading	100.0	82.0	96.0	78.2	71.2	82.2
SD2.1	T2SMark	100.0	80.1	94.7	75.3	65.3	79.9
SD2.1	PRC	100.0	79.7	78.5	74.7	55.8	76.8
SDXL	RingID	100.0	82.9	98.1	88.6	85.2	86.3
SDXL	Gauss. Shading	100.0	80.7	95.8	75.3	66.0	80.3
SDXL	T2SMark	100.0	80.2	95.5	75.2	66.6	80.1
SDXL	PRC	100.0	78.8	23.6	75.1	15.5	66.6
FLUX.1	RingID	100.0	79.8	45.0	74.8	36.9	71.3
FLUX.1	Gauss. Shading	100.0	80.1	91.9	75.3	65.7	79.6
FLUX.1	T2SMark	100.0	80.0	38.4	75.0	27.0	69.9
FLUX.1	PRC	100.0	65.8	2.8	33.8	1.5	47.6

Table 16: TPR@1%FPR (%) across the eight content subclasses, pooled over methods within each watermark family (post-hoc methods in identity mode).

Image category	Attack category	portrait	people	sports	indoor	food	vehicle	landscape	object
fake	C1	82.1	81.9	81.8	82.0	81.9	81.8	81.4	82.1
fake	C2	44.5	45.4	45.2	45.2	47.3	44.7	44.7	44.7
fake	C3	71.2	70.9	71.2	71.0	71.4	71.0	70.8	71.5
fake	C4	39.5	41.1	41.2	40.6	43.2	40.5	40.8	40.4
real	C1	81.5	81.7	81.1	81.4	81.1	81.4	80.6	81.5
real	C2	42.2	42.9	41.3	42.4	42.4	41.3	42.1	42.3
real	C3	69.1	69.6	68.4	68.3	67.8	68.5	68.8	68.9
real	C4	38.1	39.1	38.4	39.1	40.2	38.1	38.9	38.7
synthetic	C1	82.8	82.7	82.6	82.6	82.6	82.5	82.5	82.7
synthetic	C2	46.2	46.9	46.9	45.1	43.7	43.7	43.6	42.0
synthetic	C3	73.4	74.0	73.3	73.4	73.2	73.3	73.8	73.1
synthetic	C4	42.9	44.2	43.7	42.6	40.3	40.4	40.8	38.5
in-gen.	C1	79.3	79.4	79.2	79.0	79.2	80.0	80.8	78.8
in-gen.	C2	69.8	74.4	69.3	71.7	70.1	73.9	72.9	68.6
in-gen.	C3	72.9	73.7	73.0	73.6	72.8	74.1	75.2	72.5
in-gen.	C4	51.6	56.2	50.9	51.5	52.2	54.3	52.7	50.4

messages. For Qwen2.5-7B, Ministral-8B, and Llama-3.1-8B, 83-92% of these incorrect answers are the reply `unknown`: the reader abstains on a short message such as "ali got ham" although the recovered text is correct. The QSR of a short-message method therefore depends on the reader as well as on the watermark, whereas the ordering of the attack categories and the difference between the two watermark families are observed under every reader.

F.6 Calibration under attack

We additionally evaluate what happens when the detection threshold is calibrated using only unattacked unwatermarked images and then held fixed under image processing.

The realized FPR exceeds 1% under at least one condition for 7-9 of the 11 post-hoc methods in each image category. The largest identity-embedded message FPRs are 3.4% for InvisMark on manipulated images, 2.8% for WAM on real images, and 5.6% for WAM on synthetic images. Thus, calibration using only unattacked negatives does not necessarily preserve the target FPR after image processing.

Table 17: Spread of TPR@1%FPR across the eight content subclasses for each method (maximum minus minimum, in percentage points), pooled over attacked conditions (post-hoc methods in identity mode).

Method	manipulated	Real	Synthetic
CIN	0.6	0.5	1.3
PIMoG	3.1	4.2	3.1
WAM	0.6	2.6	1.2
TrustMark	4.5	6.7	6.2
EditGuard	2.5	2.4	2.8
VINE	1.3	1.1	0.9
InvisMark	3.1	5.8	5.1
OmniGuard	1.1	0.8	2.9
MBRS	1.7	1.6	1.7
VideoSeal	2.2	2.1	2.0
PixelSeal	1.0	1.2	1.8
RingID	–	–	2.0
Gauss. Shading	–	–	1.2
T2SMark	–	–	1.4
PRC	–	–	6.7

Table 18: Message-level results on the real and manipulated categories and on all 1,593 synthetic images for post-hoc methods. Class is the message length class. The columns under QSR report Question Success Rate before attack (Unattacked), in the four attack categories (C1-C4), and over all attacked conditions (All). sem-AURSC and bit-AURSC are the normalized areas under the QSR-versus-ASI and bit-accuracy-versus-ASI curves over $[0, 0.2]$, computed over all attacks. Msg. Rec. and Bit Acc. are mean complete-message recovery and mean bit accuracy over all attacked conditions.

		QSR									
Method	Class	Unattacked	C1	C2	C3	C4	All	sem-AURSC	bit-AURSC	Msg. Rec.	Bit Acc.
post-hoc, real category											
EditGuard	short	0.861	0.369	0.001	0.138	0.000	0.241	0.578	0.867	0.257	0.669
VINE	short	0.857	0.661	0.278	0.621	0.263	0.558	0.752	0.958	0.603	0.870
InvisMark	short	0.219	0.127	0.008	0.030	0.000	0.081	0.152	0.832	0.016	0.732
OmniGuard	short	0.871	0.646	0.346	0.691	0.364	0.581	0.754	0.963	0.624	0.893
MBRS	long	0.978	0.517	0.104	0.267	0.043	0.369	0.679	0.911	0.244	0.790
VideoSeal	long	0.960	0.693	0.146	0.724	0.134	0.561	0.799	0.963	0.382	0.895
PixelSeal	long	0.968	0.740	0.142	0.727	0.107	0.586	0.803	0.958	0.422	0.895
post-hoc, fake category											
EditGuard	short	0.862	0.362	0.001	0.136	0.000	0.235	0.554	0.853	0.253	0.666
VINE	short	0.861	0.673	0.335	0.626	0.292	0.576	0.750	0.958	0.631	0.877
InvisMark	short	0.233	0.146	0.010	0.037	0.000	0.093	0.153	0.829	0.018	0.753
OmniGuard	short	0.872	0.639	0.334	0.685	0.354	0.571	0.746	0.962	0.620	0.892
MBRS	long	0.980	0.517	0.107	0.267	0.049	0.368	0.676	0.909	0.239	0.789
VideoSeal	long	0.969	0.741	0.155	0.822	0.172	0.609	0.811	0.962	0.452	0.903
PixelSeal	long	0.973	0.790	0.148	0.874	0.154	0.641	0.818	0.958	0.494	0.903
post-hoc, synthetic category (all 1,593 images)											
EditGuard	short	0.871	0.388	0.002	0.173	0.000	0.256	0.567	0.846	0.279	0.669
VINE	short	0.863	0.682	0.414	0.638	0.387	0.605	0.772	0.966	0.661	0.892
InvisMark	short	0.240	0.157	0.008	0.068	0.001	0.104	0.171	0.862	0.020	0.783
OmniGuard	short	0.871	0.644	0.304	0.694	0.319	0.567	0.741	0.959	0.605	0.892
MBRS	long	0.980	0.520	0.116	0.290	0.060	0.376	0.663	0.901	0.233	0.789
VideoSeal	long	0.974	0.762	0.120	0.831	0.123	0.611	0.823	0.962	0.455	0.902
PixelSeal	long	0.975	0.804	0.111	0.836	0.096	0.633	0.820	0.957	0.493	0.905

G REPRODUCTION

The released toolchain includes code for image generation with six generators, watermark embedding with four in-generation and 11 post-hoc methods, attack execution using fixed processing-chain specifications, watermark decoding, and computation of all reported metrics. All commands and

Table 19: QSR, complete-message recovery (Msg. Rec.), and bit accuracy (Bit Acc.) by attack category (C1-C4; All is the mean over all attacked conditions) for each watermark family and image category, under the primary reader. Values are averaged over the 7 post-hoc or the 3 in-generation methods that carry sentence embedded messages. The three metrics are computed from the same recovered embedded messages. Synthetic (matched) rows use the 1,193-image cohort of Table 4; synthetic (all 1,593) rows use the full synthetic subset.

	Unattacked	C1	C2	C3	C4	All
QSR						
post-hoc, fake	0.821	0.552	0.156	0.492	0.146	0.442
post-hoc, real	0.816	0.536	0.146	0.457	0.130	0.425
post-hoc, synthetic (all 1,593)	0.825	0.565	0.153	0.504	0.141	0.450
post-hoc, synthetic (matched)	0.824	0.564	0.154	0.503	0.142	0.454
in-generation, synthetic (matched)	0.984	0.737	0.354	0.647	0.236	0.635
Msg. Rec.						
post-hoc, fake	0.845	0.498	0.114	0.412	0.105	0.387
post-hoc, real	0.836	0.475	0.106	0.357	0.097	0.364
post-hoc, synthetic (all 1,593)	0.852	0.508	0.111	0.415	0.106	0.392
post-hoc, synthetic (matched)	0.852	0.508	0.112	0.414	0.106	0.396
in-generation, synthetic (matched)	1.000	0.737	0.236	0.627	0.148	0.610
Bit Acc.						
post-hoc, fake	0.990	0.878	0.700	0.846	0.683	0.826
post-hoc, real	0.989	0.873	0.691	0.836	0.676	0.821
post-hoc, synthetic (all 1,593)	0.991	0.884	0.703	0.863	0.689	0.833
post-hoc, synthetic (matched)	0.991	0.883	0.704	0.862	0.692	0.835
in-generation, synthetic (matched)	1.000	0.908	0.838	0.879	0.762	0.883

Table 20: Partial-recovery analysis under the primary reader. The population is every attacked image of the methods that carry sentence embedded messages, over all attacked conditions. For each method, images are pooled over the attacked conditions, the synthetic post-hoc evaluation subsets are combined with image-count weights, and the table reports the mean over methods. Msg. Rec. is complete-message recovery and QSR is computed over all messages. QSR (exact) and QSR (not exact) are computed on messages that were and were not recovered exactly. Share is the percentage of all correct answers that come from messages that were not recovered exactly. The synthetic (matched) row uses the 1,193-image cohort.

	Msg. Rec.	QSR	QSR (exact)	QSR (not exact)	Share (%)
post-hoc, fake	0.387	0.442	0.914	0.152	27.1
post-hoc, real	0.364	0.425	0.912	0.148	28.6
post-hoc, synthetic (all 1,593)	0.392	0.450	0.900	0.153	27.5
post-hoc, synthetic (matched)	0.396	0.454	0.891	0.155	27.3
in-generation, synthetic (matched)	0.610	0.635	0.984	0.098	5.1

random seeds are documented. A single script generates the tables from the released metric files, and the figures use the same files.

Table 21: Reader sensitivity of QSR. Mean QSR over all attacked conditions under five frozen readers, averaged over the 7 post-hoc and 3 in-generation methods that carry sentence embedded messages. The last row is the difference between the highest and the lowest reader. The matched columns use the 1,193-image cohort of Table 4. Per-attack-category and per-method results are reported in Tables 23 and 24.

	Post-hoc				In-generation
Reader	manipulated	Real	Synthetic (all 1,593)	Synthetic (matched)	Synthetic (matched)
Qwen2.5-7B (primary)	0.442	0.425	0.450	0.454	0.635
Qwen3.5-9B	0.495	0.476	0.503	0.507	0.647
Ministral-8B	0.434	0.417	0.441	0.446	0.623
Gemma-3-12B	0.478	0.462	0.486	0.490	0.642
Llama-3.1-8B	0.417	0.397	0.422	0.427	0.627
Max − min	0.078	0.079	0.080	0.080	0.024

Table 22: Agreement of each additional reader with the primary reader, Qwen2.5-7B-Instruct. Pearson's r, Spearman's ρ, and the mean absolute difference are computed on condition-level QSR over 2,385 method-attack pairs. Agreement is the percentage of the 7.24 million question instances on which both readers are correct or both are incorrect, and κ is Cohen's kappa for the same comparison. The last column is Pearson's r of condition-level ΔNLL.

	Condition-level QSR			Per question		ΔNLL
Reader	r	ρ	Mean $\lvert\Delta\rvert$	Agreement (%)	κ	r
Qwen3.5-9B	0.994	0.970	0.050	94.2	0.884	0.997
Ministral-8B	1.000	0.996	0.010	93.8	0.877	0.994
Gemma-3-12B	0.997	0.989	0.034	94.2	0.884	0.990
Llama-3.1-8B	0.998	0.992	0.026	93.3	0.867	0.997

Table 23: Mean QSR by attack category under each reader, for each watermark family and image category. Values are averaged over the 7 post-hoc or the 3 in-generation methods that carry sentence embedded messages.

Reader	C1	C2	C3	C4	All
post-hoc, manipulated					
Qwen2.5-7B (primary)	0.552	0.156	0.492	0.146	0.442
Qwen3.5-9B	0.615	0.191	0.544	0.178	0.495
Ministral-8B	0.543	0.155	0.484	0.143	0.434
Gemma-3-12B	0.595	0.181	0.527	0.167	0.478
Llama-3.1-8B	0.521	0.145	0.467	0.135	0.417
post-hoc, real					
Qwen2.5-7B (primary)	0.536	0.146	0.457	0.130	0.425
Qwen3.5-9B	0.596	0.177	0.509	0.159	0.476
Ministral-8B	0.526	0.145	0.446	0.128	0.417
Gemma-3-12B	0.579	0.170	0.494	0.151	0.462
Llama-3.1-8B	0.501	0.135	0.427	0.119	0.397
post-hoc, synthetic (all 1,593)					
Qwen2.5-7B (primary)	0.565	0.153	0.504	0.141	0.450
Qwen3.5-9B	0.626	0.187	0.557	0.171	0.503
Ministral-8B	0.554	0.151	0.493	0.137	0.441
Gemma-3-12B	0.606	0.178	0.540	0.162	0.486
Llama-3.1-8B	0.531	0.142	0.475	0.129	0.422
post-hoc, synthetic (matched)					
Qwen2.5-7B (primary)	0.564	0.154	0.503	0.142	0.454
Qwen3.5-9B	0.626	0.188	0.557	0.174	0.507
Ministral-8B	0.554	0.152	0.493	0.139	0.446
Gemma-3-12B	0.606	0.179	0.539	0.164	0.490
Llama-3.1-8B	0.532	0.143	0.475	0.131	0.427
in-generation, synthetic (matched)					
Qwen2.5-7B (primary)	0.737	0.354	0.647	0.236	0.635
Qwen3.5-9B	0.747	0.376	0.656	0.251	0.647
Ministral-8B	0.725	0.339	0.636	0.227	0.623
Gemma-3-12B	0.743	0.367	0.653	0.245	0.642
Llama-3.1-8B	0.729	0.345	0.640	0.230	0.627

Table 24: QSR of each method under each reader, averaged over all attacked conditions.

Method	Qwen2.5-7B	Qwen3.5-9B	Ministral-8B	Gemma-3-12B	Llama-3.1-8B
post-hoc, synthetic (matched)					
EditGuard	0.261	0.306	0.259	0.291	0.237
VINE	0.604	0.702	0.602	0.668	0.546
InvisMark	0.104	0.165	0.102	0.149	0.094
OmniGuard	0.575	0.673	0.573	0.641	0.522
MBRS	0.389	0.414	0.373	0.406	0.376
VideoSeal	0.614	0.639	0.599	0.631	0.600
PixelSeal	0.629	0.653	0.614	0.646	0.616
in-generation, synthetic (matched)					
Gauss. Shading	0.685	0.700	0.670	0.694	0.675
T2SMark	0.693	0.706	0.680	0.701	0.684
PRC	0.527	0.533	0.518	0.531	0.521
post-hoc, fake					
EditGuard	0.235	0.280	0.235	0.266	0.212
VINE	0.576	0.677	0.577	0.641	0.518
InvisMark	0.093	0.146	0.087	0.131	0.081
OmniGuard	0.571	0.672	0.570	0.638	0.514
MBRS	0.368	0.392	0.352	0.385	0.359
VideoSeal	0.609	0.635	0.593	0.628	0.600
PixelSeal	0.641	0.665	0.626	0.660	0.633
post-hoc, real					
EditGuard	0.241	0.284	0.241	0.272	0.215
VINE	0.558	0.649	0.558	0.622	0.498
InvisMark	0.081	0.128	0.077	0.115	0.070
OmniGuard	0.581	0.678	0.581	0.650	0.519
MBRS	0.369	0.394	0.353	0.387	0.357
VideoSeal	0.561	0.589	0.542	0.582	0.548
PixelSeal	0.586	0.613	0.567	0.607	0.572
post-hoc, synthetic (all 1,593)					
EditGuard	0.256	0.299	0.253	0.285	0.230
VINE	0.605	0.700	0.599	0.666	0.542
InvisMark	0.104	0.164	0.101	0.149	0.092
OmniGuard	0.567	0.662	0.561	0.631	0.509
MBRS	0.376	0.400	0.359	0.392	0.364
VideoSeal	0.611	0.636	0.596	0.628	0.599
PixelSeal	0.633	0.656	0.617	0.649	0.621

Table 25: Calibration gap on the identity-embedded message evaluation subsets. For each method and category, the table reports the false-positive rate (%) of a nominal 1% threshold fitted on unattacked negatives and evaluated on unattacked negatives (unwatermarked), all attacked-unwatermarked negatives (att.), and the condition with the highest realized FPR (max).

	Manipulated			Real			Synthetic		
Method	unwatermarked	att.	max	unwatermarked	att.	max	unwatermarked	att.	max
CIN	0.3	0.3	0.6 (rotation 10)	0.3	0.2	0.8 (UnMarker)	0.3	0.3	0.9 (JPEG 50)
PIMoG	0.3	0.3	0.6 (JPEG 50)	0.2	0.3	0.6 (crop 0.5)	0.4	0.3	1.1 (brightness 0.7)
WAM	0.6	0.8	2.0 (JPEG 30)	0.7	1.0	2.8 (JPEG 30)	0.6	1.2	5.6 (JPEG 30)
TrustMark	0.8	0.8	1.4 (crop 0.5)	0.8	0.9	1.4 (crop 0.7)	0.8	0.8	1.4 (brightness 1.5)
EditGuard	0.3	0.4	0.8 (brightness 1.2)	0.8	1.1	1.9 (edit FLUX.2)	0.4	0.4	0.9 (crop 0.5)
VINE	0.9	0.9	1.8 (unmarkshare)	0.4	0.4	0.7 (filterpost)	0.7	0.6	1.7 (social)
InvisMark	0.5	0.8	3.4 (vaeregen)	0.4	0.4	1.2 (JPEG 30)	0.6	1.4	5.0 (vaeregen)
OmniGuard	0.5	0.4	0.7 (blur 5)	0.7	0.5	1.1 (JPEG 50)	0.8	0.9	1.9 (brightness 1.5)
MBRS	1.0	1.1	1.7 (JPEG 70)	0.8	1.0	1.5 (resize 0.5)	0.8	0.8	1.4 (contrast 1.2)
VideoSeal	0.6	0.7	1.1 (brightness 1.2)	0.6	0.6	1.1 (straighten)	0.8	0.9	1.7 (brightness 1.2)
PixelSeal	1.0	1.0	1.6 (contrast 0.7)	0.6	0.6	1.2 (messenger)	0.8	0.7	1.5 (filterpost)

Table 26: Synthetic post-hoc evaluation subsets listed separately for identity embedded messages. Open-set attribution accuracy at 1% FPR is reported by attack category, together with AURSC and Obs. Min@0.2 on TPR@1%FPR. Images is the number of images in the subset. Unattacked is the result before attack, C1-C4 are the four attack categories, and All is the mean over all attacked conditions. Category-level synthetic results pool these subsets.

Subset	Method	Images	Unattacked	C1	C2	C3	C4	All	AURSC	Obs. Min@0.2
SD2.1/SDXL/FLUX.1	CIN	796	100.0	70.1	11.9	19.2	0.1	45.5	0.676	0.002
SD2.1/SDXL/FLUX.1	PIMoG	796	100.0	72.0	41.1	48.2	24.9	58.4	0.786	0.004
SD2.1/SDXL/FLUX.1	WAM	796	100.0	96.9	61.5	99.6	63.9	88.2	0.968	0.771
SD2.1/SDXL/FLUX.1	TrustMark	796	99.9	77.5	20.4	83.7	22.4	63.6	0.869	0.008
SD2.1/SDXL/FLUX.1	EditGuard	796	100.0	61.3	7.1	30.1	0.8	41.4	0.728	0.000
SD2.1/SDXL/FLUX.1	VINE	796	100.0	80.0	93.6	75.0	71.2	80.4	0.940	0.973
SD2.1/SDXL/FLUX.1	InvisMark	796	100.0	93.0	18.3	86.7	15.1	71.9	0.885	0.000
SD2.1/SDXL/FLUX.1	OmniGuard	796	100.0	82.3	77.3	87.5	67.3	80.7	0.933	0.157
SD2.1/SDXL/FLUX.1	MBRS	796	100.0	79.7	35.9	75.0	46.9	68.6	0.860	0.001
SD2.1/SDXL/FLUX.1	VideoSeal	796	100.0	94.8	54.1	100.0	57.8	85.3	0.953	0.014
SD2.1/SDXL/FLUX.1	PixelSeal	796	100.0	99.7	41.6	100.0	55.5	85.9	0.946	0.010
six generators	CIN	797	100.0	70.0	12.3	17.9	0.1	46.2	0.688	0.000
six generators	PIMoG	797	100.0	72.4	40.1	48.6	24.0	59.0	0.792	0.010
six generators	WAM	797	100.0	97.0	65.7	99.9	68.5	90.0	0.967	0.857
six generators	TrustMark	797	99.9	76.6	23.9	82.4	28.1	64.9	0.858	0.005
six generators	EditGuard	797	100.0	60.5	5.6	27.8	0.6	41.2	0.699	0.000
six generators	VINE	797	100.0	80.0	93.3	75.0	71.7	80.3	0.944	0.971
six generators	InvisMark	797	100.0	94.0	24.2	93.2	21.8	76.3	0.894	0.003
six generators	OmniGuard	797	100.0	82.5	81.2	87.5	71.1	81.9	0.942	0.685
six generators	MBRS	797	100.0	79.6	39.3	74.9	50.9	70.1	0.868	0.005
six generators	VideoSeal	797	100.0	94.6	61.6	100.0	60.4	87.2	0.953	0.013
six generators	PixelSeal	797	100.0	99.8	44.5	100.0	57.3	87.5	0.943	0.020