MADE TO MEASURE: DESIGNING IMAGE WATERMARKS TO SPECIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Image watermarking supports provenance and attribution by embedding verifiable identity information into images. Practical deployments, however, must jointly satisfy requirements for attack resistance, false-positive rate (FPR), image quality, and latency. Existing watermarking methods are robust to different classes of transformations, so combining complementary methods can provide broader protection than any single watermark. Such composition is challenging, as additional fragments increase distortion and decoding cost and must share the same FPR budget. Therefore, we propose TAILOR, a request-conditioned watermark composition framework with three stages: (1) *offline characterization* measures fragment recovery, distortion, and runtime as response curves over embedding strength; (2) *joint configuration selection* encodes the request as an SMT model over these curves and solves for the lowest-distortion composition of fragments, strengths, order, and geometric recovery; and (3) *live calibration* validates the selected configuration on the user's images and refines predictions that fail to transfer. Experimental results across 7,321 distinct requests spanning five scenarios and 20 attack settings show that TAILOR achieves 96.21% scenario-averaged request satisfaction with a mean PSNR of 41.02 dB, outperforming existing methods in robustness while achieving consistently better image quality. Code is available at `https://anonymous.4open.science/r/Tailor-AC7E`.

1 INTRODUCTION

The growing use of advanced generative image models such as *GPT-Image-2* (OpenAI, 2026), and *Nano Banana 2* (Google DeepMind, 2026) has made provenance and attribution important concerns for creators and service providers. Invisible image watermarking associates an image with a machine-readable signal that can be checked after distribution (Bui et al., 2023; Lu et al., 2025), an approach that has already seen commercial deployment and demonstrated significant practical value. For example, Google introduced SynthID for Imagen customers on Vertex AI to help identify AI-generated images (Gowal & Kohli, 2023), while Meta introduced invisible watermarking for images generated by Meta AI to improve transparency and traceability (Meta, 2023). These deployments demonstrate the practical role of watermarking in provenance and attribution.

Existing watermarking methods improve robustness through different designs. TrustMark learns to balance image quality and watermark recovery under image perturbations (Bui et al., 2023). VINE leverages generative priors and surrogate distortions to improve robustness against image editing (Lu et al., 2025), while VideoSeal targets efficient watermarking with robustness to geometric transformations and video compression (Fernandez et al., 2024). Their designs therefore provide complementary protection against different transformations, while selecting a single watermark or tuning its strength cannot exploit the protection offered by the others. Combining multiple watermark mechanisms is thus attractive for achieving broader robustness.

Practical deployment, however, requires more than robustness alone. A watermark configuration must survive expected transformations while preserving image quality, satisfying a false-positive-rate (FPR) requirement, and meeting runtime constraints. For example, SynthID is designed to remain detectable after common image modifications while preserving image quality (Gowal & Kohli, 2023), and the SynthID-Image report further highlights low encoding overhead, high-throughput decoding, and reliable decisions at low FPRs for large-scale deployment (Gowal et al., 2025). Beyond watermark detection, C2PA provides a standardized framework for cryptographically verifiable provenance across content workflows (Coalition for Content Provenance and Authenticity, 2024),

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.

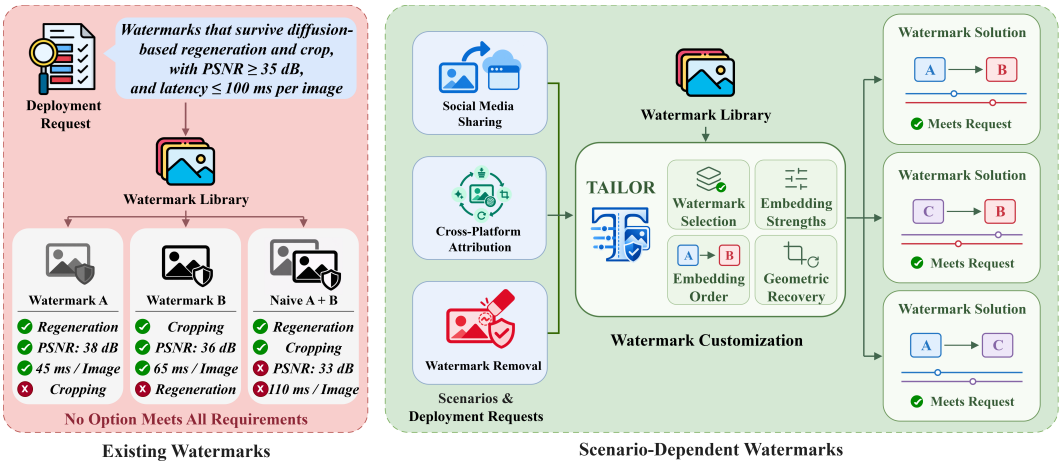


Figure 1: **Illustration of TAILOR.** Existing watermark schemes may fail to jointly satisfy scenario-specific robustness, quality, and latency requirements, while TAILOR composes and configures complementary watermarks to produce deployment-specific solutions.

while Adobe Content Authenticity combines invisible watermarking with Content Credentials and digital fingerprinting to preserve attribution after transformations such as metadata removal and screenshots (Parsons, 2024; Adobe, 2025). Deployment and access conditions can further change the relevant attack surface (Gowal et al., 2025). When multiple watermarks are combined, their fragments must share the same quality, latency, and FPR budgets, while their embedding strengths and order jointly affect robustness and distortion. Moreover, offline measurements may not transfer directly to the user's images. These coupled requirements motivate treating the expected attacks, FPR budget, image-quality floor, and runtime limit as deployment-specific inputs. The challenge is therefore to jointly select and validate a watermark composition that satisfies the complete request.

To address these challenges, we propose TAILOR, a request-conditioned framework for composing and validating image watermarks (Figure 1). The core insight is to treat existing watermark encoder and decoder pairs as complementary *fragments* whose configurations should be optimized jointly rather than selected independently. TAILOR first characterizes fragment recovery, distortion, runtime, and geometric recovery offline and represents these measurements as continuous response models. Given a request specifying attacks, FPR, image quality, and latency, an SMT-based solver (Barrett & Tinelli, 2018) jointly selects the fragments, embedding strengths, order, and geometric recovery stages that satisfy the complete specification while minimizing predicted distortion. Live calibration then evaluates candidate compositions on the user's images and refines inaccurate offline predictions when necessary. We conduct a comprehensive evaluation of TAILOR on 7,321 distinct deployment requests spanning five scenarios and 20 attack settings. TAILOR achieves 96.21% scenario-averaged request satisfaction with a mean PSNR of 41.02 dB, outperforming existing methods in robustness while achieving consistently better image quality (Table 1). Controlled ablations further validate the contributions of candidate search, watermark composition, and strength adaptation. Our contributions are summarized as follows:

- We formulate request-conditioned watermark deployment and introduce TAILOR. Given an attack set, an FPR budget, a quality floor, and a latency ceiling, the task is to construct a watermark configuration satisfying the complete specification rather than select a fixed watermark in isolation.
- We develop a configuration solver that integrates measured performance models with live calibration. SMT jointly optimizes fragment choices and settings under the requested quality and latency budgets and configuration-dependent verification thresholds, returning either a valid configuration or reporting that no configuration found within the encoded model and the candidate budget.
- We conduct extensive evaluations on 7,321 deployment requests spanning five scenarios and 20 attack settings. TAILOR achieves 96.21% scenario-averaged request satisfaction, outperforming the greedy baseline by 5.16 percentage points and Enumeration + Geo. by 13.00 points, while controlled ablations validate the benefits of candidate search, watermark composition, and strength adaptation.

2 Related Work

Image Watermarking and Attacks. Post-hoc watermarking uses transform-domain methods based on DWT with DCT (Al-Haj, 2007) and DWT with DCT and SVD (Navas et al., 2008), or learned encoder and decoder models such as RivaGAN (Zhang et al., 2019), StegaStamp (Tancik et al., 2020), WAM (Sander et al., 2025), TrustMark (Bui et al., 2023), and VINE (Lu et al., 2025). These methods face signal-processing attacks such as JPEG compression, blur, and noise; geometric attacks such as cropping, resizing, and rotation (An et al., 2024); adaptive and optimization-based removal attacks such as WEvade (Jiang et al., 2023) and UnMarker (Kassis & Hengartner, 2025); and regeneration attacks such as Regen (Zhao et al., 2024b) and CtrlRegen+ (Liu et al., 2025). Regeneration attacks reconstruct perceptually similar images while suppressing watermark signals (Zhao et al., 2024b; Liu et al., 2025). RAVEN (Shamshad et al., 2026) further exploits diffusion-based novel-view synthesis for watermark removal. Different watermarks are effective against different attacks, while no single method is uniformly robust across the evaluated threat landscape (An et al., 2024). In our evaluation, VINE better withstands regeneration but is vulnerable to the tested geometric attacks, whereas TrustMark shows the opposite pattern. This complementarity motivates combining watermarks, although sequential embedding can introduce interference and image distortion, and additional decoder executions increase latency.

Scenario-Dependent Watermark Configuration. Practical deployments impose requirements beyond average watermark recovery. Adobe Content Authenticity combines invisible watermarking with Content Credentials and digital fingerprinting for provenance across transformations such as metadata removal and screenshots (Parsons, 2024; Adobe, 2025), while SynthID emphasizes image-quality preservation, robustness to common edits, and efficient detection (Gowal & Kohli, 2023; Gowal et al., 2025). OmniGuard (Zhang et al., 2025) similarly combines copyright protection with manipulation localization, illustrating that watermarking objectives vary by application. FLEX-Mark (Nicholas et al., 2026) further argues for evaluation under shared distortion budgets, realistic payloads, low false-positive operating points, and adaptive removal rather than average recovery alone. These works motivate configuration at the system level: a social-media deployment may prioritize crop and compression robustness, while another may prioritize regeneration resistance, attribution reliability, payload capacity, image quality, or latency. Existing work develops stronger individual watermarks, broader attack benchmarks, fixed multi-purpose architectures, and watermark ensembles (Petrov et al., 2026), but to our knowledge does not directly treat deployment as a per-request constrained optimization over heterogeneous watermark fragments, their order, strengths, and recovery mechanisms. We therefore propose TAILOR, a request-conditioned framework that jointly selects watermark fragments, embedding order, strengths, and geometric recovery support under user-specified robustness, false-positive, image-quality, and latency constraints.

3 Threat Model

We consider two parties: a *protector*, who deploys and verifies the watermark, and an *attacker*, who attempts to remove it from a released watermarked image.

Protector's Goal and Capabilities. The protector specifies a deployment request $u = (\mathcal{A}_u, \alpha, q, t)$, where $\mathcal{A}_u$ is the set of attacks the watermark is expected to withstand, α is the target false-positive rate (FPR), q is the minimum image-quality requirement, and t is the deployment latency ceiling. Let x_o denote an original image and let $w \in \{0,1\}^n$ denote the keyed watermark payload. A watermark encoder $E(\cdot,\cdot)$ embeds w into x_o to produce $x_w = E(x_o, w)$. An attack $a \in \mathcal{A}_u$ transforms the released watermarked image into $\tilde{x}_w^a = a(x_w)$. The corresponding decoder $D(\cdot)$ then produces a watermark readout $w'^a = D(\tilde{x}_w^a)$. The protector seeks to preserve reliable agreement between w'^a and w under the requested attacks while satisfying the FPR, image-quality, and latency requirements in u.

We focus on *post-hoc* watermarking: the protector may modify an image after it has been generated or acquired, but does not modify the underlying image generator or its training procedure. Verification is reference-free: given a candidate image, a specified identity, the secret key, and the deployed configuration, the verifier determines whether the image carries that identity without requiring the original unwatermarked image.

Attacker's Goal and Capabilities. The attacker aims to transform the released watermarked image x_w so that its embedded identity can no longer be reliably recovered. We assume the attacker

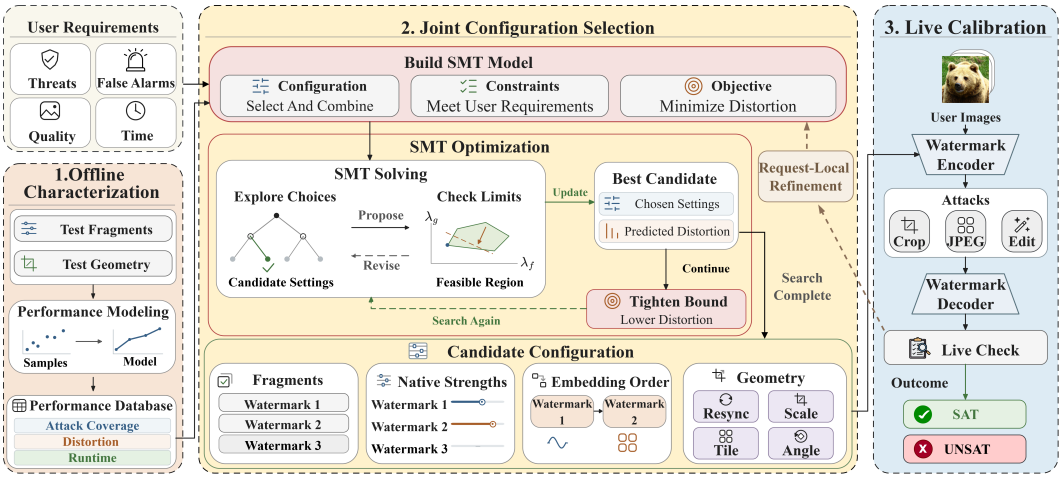


Figure 2: **Overview of TAILOR.** (1) Offline characterization builds a performance database of attack coverage, distortion, and runtime. (2) Joint configuration selection solves an SMT model of the user requirements to find the lowest-distortion configuration of fragments, strengths, order, and geometry. (3) Live calibration validates this configuration on user images under the requested attacks, returning SAT or UNSAT.

has access to x_w, but not the protector's secret key or the original unwatermarked image x_o. Depending on the requested attack, the attacker may apply signal-processing or re-encoding operations, geometric transformations, neural compression, publicly available regeneration or editing models, or optimization-based watermark-removal attacks. For each request, the attacker is granted only the capabilities required by the attacks in $\mathcal{A}_u$, and attacks are evaluated separately unless an attack is explicitly defined as a composition of multiple operations. $\mathcal{A}_u$ is the specification: the configuration TAILOR returns is certified against exactly these operators at exactly the settings the request names, which Table 6 lists.

4 METHODOLOGY

In this section, we introduce TAILOR, a request-conditioned watermark composition framework. Given a deployment request $u = (\mathcal{A}_u, \alpha, q, t)$, TAILOR constructs a watermark configuration that withstands every attack in $\mathcal{A}_u$ at false-positive rate α, with image quality at least q and latency at most t. It proceeds in three stages. First, *Offline Characterization* profiles each watermark fragment and geometric recovery stage and stores the results in a performance database. Second, *Joint Configuration Selection* encodes the request as an SMT model over this database and solves for the lowest-distortion configuration. Third, *Live Calibration* validates the resulting candidate on the user's images under the requested attacks before deployment. Figure 2 provides an overview of TAILOR.

4.1 OFFLINE CHARACTERIZATION

Measurements. Offline characterization builds the *Performance Database* used to predict candidate configurations before live calibration. For each fragment $f \in \{\text{VINE}, \text{TrustMark}, \text{VideoSeal}\}$, we sweep its native embedding strength λ_f and measure, under every attack a, the mean bit accuracy $b_f(\lambda_f, a)$ together with the underlying per-image scores. We also record standalone embedding distortion $D_f(\lambda_f)$ as mean squared error (MSE), and embedding and decoding latency. To characterize composition, $\delta_{g \to f}(\lambda_g, a)$ measures the recovery loss of f when g is embedded after it, while e_{fg} measures excess pairwise distortion beyond the standalone terms. We repeat the relevant measurements for the four optional geometric recovery stages derived from the fragments' geometric failure modes: scale, tile, resynchronization, and angle search (Appendix B). All offline measurements use a fixed image subset disjoint from the images later used for live calibration (Appendix C.1).

Continuous Strengths. The database observes performance only at discrete measured strengths, whereas configuration selection optimizes strength continuously. We therefore represent each measured response with a piecewise-linear (PWL) model. Given measurements $y^{(k)}$ at knots

$\lambda^{(0)} < \cdots < \lambda^{(K)}$, we define

$$\widehat{y}(\lambda) = y^{(k)} + s^{(k)}\big(\lambda - \lambda^{(k)}\big), \qquad s^{(k)} = \frac{y^{(k+1)} - y^{(k)}}{\lambda^{(k+1)} - \lambda^{(k)}}, \qquad \lambda \in \big[\lambda^{(k)}, \lambda^{(k+1)}\big]. \tag{1}$$

This interpolation reproduces the measured values while providing predictions for any strength within the supported range, allowing λ_f to be optimized continuously. Because each segment is linear, the curves can also be encoded directly in the SMT model (Appendix A.1).

Predicted Quantities. For a candidate configuration c, predicted recovery combines each fragment's standalone response with the measured effects of fragments embedded after it:

$$\widehat{b}_f(c, a) = b_f(\lambda_f, a) - \sum_{g \in S:\, f \prec_\pi g} \delta_{g \to f}(\lambda_g, a), \tag{2}$$

where the corresponding measured geometric response is used when the selected recovery stage applies to fragment f and attack a.

We model distortion in MSE rather than directly in PSNR for two reasons. Sequential embedding approximately composes distortion in the pixel-error domain, with deviations from additivity captured by the measured pairwise corrections, whereas PSNR values are not additive under composition. Moreover, modeling distortion in MSE keeps the resulting constraints linear, whereas directly modeling PSNR would introduce a logarithmic term into the solver. We therefore define

$$D(c) = \sum_{f \in S} D_f(\lambda_f) + \sum_{\{f,g\} \subseteq S} e_{fg}(\lambda_f + \lambda_g) + D_\varphi(c), \tag{3}$$

and enforce the requested PSNR floor through the equivalent MSE constraint. Since $P(c) = 10 \log_{10}\big(255^2 / D(c)\big)$, the requirement $P(c) \geq q$ is equivalent to $D(c) \leq 255^2\, 10^{-q/10}$, and minimizing $D(c)$ is therefore equivalent to maximizing PSNR. We use the PSNR form for reporting.

Predicted latency $L(c)$ combines the embedding and decoding costs of the selected fragments with the selected stage overhead. The stored per-image scores additionally estimate the fraction of images exceeding a verification threshold, from which $\Gamma_{f,a}(c)$ determines whether fragment f satisfies the required image-level acceptance rate under attack a. Thus, configuration selection considers both mean recovery and per-image reliability. Additional database construction details are provided in Appendix A.2.

4.2 Joint Configuration Selection

Given a deployment request $u = (\mathcal{A}_u, \alpha, q, t)$, TAILOR must jointly decide which watermark fragments to use, how strongly and in what order to embed them, and whether to enable geometric recovery. These choices are coupled: adding a fragment may improve attack coverage but also increase distortion, latency, and the number of verification paths sharing the FPR budget. We therefore formulate configuration selection as a joint constrained optimization problem.

Configuration. We represent a configuration as $c = (S, \lambda, \pi, \varphi)$, where S is the selected fragment subset, λ the native embedding strengths, $\pi = (f_1, \ldots, f_{|S|})$ the embedding order, and φ the enabled geometric recovery stage, of which at most one may be selected. The configuration induces a composite encoder E_c that applies the selected fragments sequentially:

$$x^{(0)} = x_o, \qquad x^{(j)} = E^{\varphi}_{f_j}\big(x^{(j-1)}, w; \lambda_{f_j}\big), \quad j = 1, \ldots, |S|. \tag{4}$$

It also determines a configured decoder D_c that evaluates each selected fragment, their fused readout, and any views introduced by the geometric recovery stage. All fragments carry the same 100-bit payload w, after fragment-specific key-derived permutation and sign masking, so composition provides multiple recovery paths for one identity rather than independent payloads. The resulting search space contains 15 ordered nonempty fragment subsets, their compatible geometric stages, and a continuous strength for every selected fragment.

Constraints. Each requirement in the deployment request becomes a constraint on c.

FPR constraint. A configuration may evaluate multiple verification paths, including individual-fragment, fused, and geometric readouts. Let $T(c)$ denote the maximum number of permitted tests. Under the null hypothesis that the image does not contain the queried watermark identity,

each aligned readout is bounded by the 100-bit random-match model $X \sim \mathrm{Bin}(100, 1/2)$ (Appendix A.3). We allocate the total FPR budget α across the $T(c)$ tests and define

$$\tau(c, \alpha) = \frac{1}{100} \min \Big\{ h \in \{0, \ldots, 101\} : \Pr[X \geq h] \leq \frac{\alpha}{T(c)} \Big\}. \tag{5}$$

A union bound then keeps the configuration-level false-positive probability within α. Thus, configurations with more verification paths must satisfy stricter recovery thresholds.

Attack coverage. Offline recovery predictions may be optimistic on unseen images, so selection uses conservative screening margins. The mean-accuracy margin η gives $\bar{\tau}(c, \alpha) = \min(1, \tau(c, \alpha) + \eta)$, while the rate margin η_{rate} raises the required image-acceptance rate from ρ to $\bar{\rho} = \min(1, \rho + \eta_{\mathrm{rate}})$. Let $\Gamma_{f,a}(c)$ indicate whether fragment f is predicted to reach $\bar{\rho}$ under attack a. We define

$$\mathrm{Cov}_{f,a}(c) \equiv \Big[\widehat{b}_f(c, a) \geq \bar{\tau}(c, \alpha) \ \wedge \ \Gamma_{f,a}(c) \Big], \qquad \mathrm{Cov}_a(c) \equiv \bigvee_{f \in S} \mathrm{Cov}_{f,a}(c). \tag{6}$$

Different fragments may therefore cover different attacks, but every requested attack must be covered by at least one selected fragment. These margins are used only for offline screening; live calibration evaluates candidates using the original $\tau(c, \alpha)$ and ρ.

Quality and latency. The predicted image quality and deployment latency must satisfy $P(c) \geq q$ and $L(c) \leq t$, respectively.

Objective. Once the request constraints define the feasible set, TAILOR selects the feasible configuration with minimum predicted distortion:

$$\min_c \ D(c) \quad \text{s.t.} \quad \Phi_u(c) \equiv \mathrm{Struct}_u(c) \wedge \bigwedge_{a \in \mathcal{A}_u} \mathrm{Cov}_a(c) \wedge P(c) \geq q \wedge L(c) \leq t, \tag{7}$$

where $\mathrm{Struct}_u(c)$ collects validity and compatibility constraints, including that every selected fragment contributes to at least one requested attack. Because PSNR decreases monotonically with MSE, minimizing $D(c)$ is equivalent to maximizing predicted PSNR among configurations that satisfy the complete request.

SMT Optimization. The remaining challenge is that the configuration space mixes discrete and continuous decisions. Boolean variables encode fragment selection, pairwise embedding order, the geometric recovery stage, and the active PWL segments from Eq. 1, while real variables encode continuous embedding strengths. SMT therefore searches fragment composition and parameter settings jointly rather than fixing a subset before tuning its strengths.

Starting from a feasible incumbent with distortion D_{best}, the solver asks whether

$$\Phi_u(c) \ \wedge \ D(c) < D_{\mathrm{best}} - \epsilon \tag{8}$$

is satisfiable for tolerance $\epsilon > 0$. A satisfying assignment replaces the incumbent, and optimization continues until no configuration improves the objective by more than ϵ. The final incumbent becomes the candidate passed to live calibration, including its selected fragments, strengths, embedding order, and geometric recovery stage.

4.3 LIVE CALIBRATION

The performance database describes the offline image pool, which may differ from the user's images. TAILOR therefore runs each candidate through the complete encoder, attack and decoder pipeline on the user's images before deployment.

Live Check. For user image $x_{o,i}$ and attack $a \in \mathcal{A}_u$, the verifier score is $s_i(c, a) = \mathrm{BA}\left(w, D_c(a(E_c(x_{o,i}, w)))\right)$, where BA denotes bit accuracy. With N_a evaluated images and required image-acceptance rate $\rho \in (0, 1]$, we require

$$\frac{1}{N_a} \sum_i \mathbf{1}\big[s_i(c, a) \geq \tau(c, \alpha) \big] \geq \rho, \qquad \frac{1}{N_a} \sum_i s_i(c, a) \geq \tau(c, \alpha), \qquad \forall a \in \mathcal{A}_u. \tag{9}$$

These conditions enforce both image-level acceptance and mean recovery. The measured PSNR and latency must also satisfy q and t.

Table 1: Request satisfaction and image fidelity across deployment scenarios.

	Request satisfaction (%) ↑						*Image fidelity*		
Method	S1	S2	S3	S4	S5	**Avg.**	PSNR ↑	SSIM ↑	PSNR-P
StegaStamp	0.00	0.00	0.00	0.00	0.00	0.00	-	-	-
MaskWM	10.32	0.00	0.00	0.00	0.00	2.06	38.51	0.9774	38.51 / 47.03
VINE	8.83	0.00	42.68	54.26	0.00	21.15	35.01	0.9931	35.01 / 40.26
TrustMark	**100.00**	0.00	23.38	0.00	0.00	24.68	40.91	0.9909	40.91 / 43.55
VideoSeal	27.98	49.74	4.01	0.00	2.74	16.89	**45.16**	**0.9949**	45.16 / 45.96
VINE + Geo.	8.83	0.00	42.68	62.65	0.00	22.83	34.87	0.9921	34.87 / 39.92
TrustMark + Geo.	**100.00**	94.71	23.38	0.00	11.94	46.01	40.08	0.9884	40.08 / 43.70
Enumeration	**100.00**	85.38	88.78	55.60	51.59	76.27	38.49	0.9902	38.49 / 41.79
Enumeration + Geo.	**100.00**	**100.00**	88.78	63.99	63.29	83.21	37.38	0.9874	37.38 / 41.41
TAILOR-G	**100.00**	97.39	87.55	76.03	94.27	91.05	40.91	0.9914	40.91 / 41.22
TAILOR-F	**100.00**	**100.00**	**97.31**	**85.40**	**98.34**	**96.21**	41.02	0.9923	-

PSNR-P: row method / TAILOR-F, evaluated on jointly accepted requests.

Request-Local Refinement. Candidates are tested in order of predicted distortion, up to three distinct discrete structures. If all fail, the measured fragment accuracies and pass rates replace the corresponding predictions for this request, and the solver proposes one revised candidate, which must pass the same live check. If no candidate passes under the current screening margins, TAILOR retries with the smaller margins of a predeclared schedule, keeping the request, the FPR accounting, and the live criteria unchanged (Appendix A.5).

Outcome. The first candidate that passes every live check is returned as c_{out}, and the outcome is SAT. The deployment then watermarks its images with $E_{c_{\text{out}}}$ and verifies them with $D_{c_{\text{out}}}$ at the same threshold $\tau(c_{\text{out}}, \alpha)$. If no candidate passes, the outcome is UNSAT: no configuration found within the encoded model and the candidate budget meets the request on the user's images.

5 EVALUATION

5.1 EXPERIMENT SETUP

Datasets and Requests. We assemble a pool of natural and generated images from MS-COCO (Lin et al., 2014), DiffusionDB (Wang et al., 2023), UltraEdit (Zhao et al., 2024a), DIV2K (Agustsson & Timofte, 2017), and LSDIR (Li et al., 2023), together with images generated by DALL·E 3 and SDXL. These sources cover diverse image characteristics, including real photographs, diffusion-generated content, editing-oriented images and multiple generative models, providing broad coverage of realistic deployment inputs. Each deployment request specifies an attack set, an FPR budget, an image-quality floor, and a latency ceiling. After sampling and deduplication, the benchmark contains 7,321 distinct requests over 20 attack settings across five scenarios: signal processing and re-encoding (S1), geometric transformations (S2), regeneration and editing (S3), watermark removal (S4), and broad attack coverage (S5). Figure 3 shows how many requests in each scenario include each attack; a request may include multiple attacks. Image sampling and attack settings are detailed in Appendices C.1 and C.2.

Baselines. We compare TAILOR with **MaskWM** (Hu et al., 2026), **StegaStamp** (Tancik et al., 2020), **VINE** (Lu et al., 2025), **TrustMark** (Bui et al., 2023), and **VideoSeal** (Fernandez et al., 2024) at unit embedding strength. For VINE and TrustMark, we additionally evaluate variants equipped with geometric recovery stages (Geo.). We also consider two composition baselines. **TAILOR-G** incrementally adds the fragment with the largest predicted coverage gain without backtracking, while reoptimizing strength, embedding order, and geometric recovery stage for each candidate subset using the same offline database and SMT model as TAILOR. **Enumeration** evaluates all seven nonempty watermark subsets at unit strength and a fixed embedding order, while **Enumeration + Geo.** additionally evaluates geometric recovery stages. All methods use the same requests, live images, and acceptance gates. Detailed baseline policies are provided in Appendix C.3.

Evaluation Metrics. We evaluate TAILOR in terms of **request satisfaction** and **image fidelity**. Request satisfaction is the percentage of distinct requests for which a configuration passes live valida-

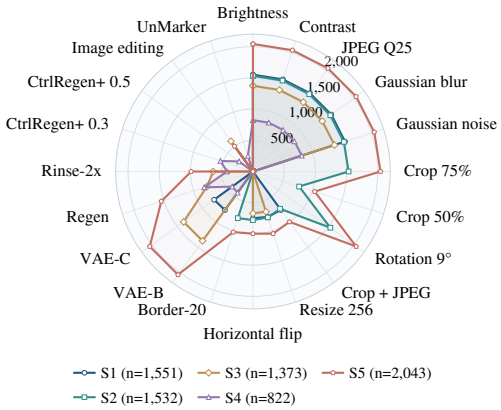


Figure 3: Number of requests containing each attack across the five scenarios.

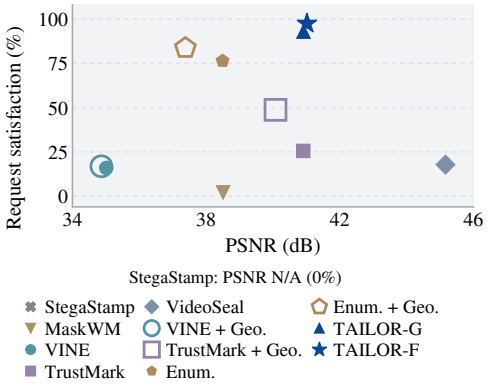


Figure 4: Overall request satisfaction and image fidelity.

tion under every requested attack while also satisfying the specified FPR, image-quality, and latency requirements. We report the satisfaction rate for each scenario and the unweighted average of these rates. Image fidelity is measured by **PSNR** (Huynh-Thu & Ghanbari, 2008) and **SSIM** (Wang et al., 2004) between the original and watermarked images before attack over each method's accepted requests. PSNR measures pixel-level fidelity, with a higher value indicating smaller changes to the original pixel values, while SSIM measures the preservation of local image structure, luminance, and contrast. Because methods may accept different sets of requests, we additionally report **PSNR-P**, which compares the mean PSNR of each method and TAILOR on their jointly accepted requests. PSNR-P separates fidelity from coverage: it holds the request set fixed and asks how much fidelity each method delivers on it.

5.2 EXPERIMENTAL RESULTS

Request Satisfaction. We evaluate whether TAILOR can satisfy complete deployment requests across different attack scenarios. As shown in Table 1, TAILOR achieves scenario-averaged satisfaction of 96.21%, compared with 91.05% for the greedy variant and 83.21% for Enumeration + Geo. The gap over greedy shows that configuration search remains important even when both methods share the same offline model, while the gap over Enumeration + Geo. shows that enumerating unit-strength compositions does not replace joint optimization of fragments and their parameters. TAILOR satisfies all S1 and S2 requests, reaches 97.31% on S3 and 98.34% on S5, and achieves 85.40% on the more challenging S4. Across all attack-conditioned request groups in Figure 5, TAILOR also achieves the highest satisfaction among the compared methods, demonstrating consistently strong coverage across the attack suite.

Image Fidelity. We evaluate the fidelity of watermarked images produced by accepted configurations. As shown in Table 1, TAILOR obtains a mean PSNR of 41.02 dB and an SSIM of 0.9923; Figure 4 illustrates its overall satisfaction and PSNR. Some single-watermark baselines achieve higher standalone PSNR, e.g., 45.16 dB for VideoSeal, because they embed only one watermark and satisfy a substantially narrower set of requests that do not require more complex compositions. These averages are therefore not directly comparable across methods. We further report **PSNR-P**, which compares TAILOR and each baseline on their jointly accepted requests, i.e., requests satisfied by both methods. On these matched requests, TAILOR achieves higher PSNR than every baseline in all completed comparisons, including gains of 0.79 dB over VideoSeal, despite VideoSeal having the highest standalone PSNR in Table 1. This shows that, under the same deployment requirements, TAILOR consistently provides better image fidelity than the baselines.

Watermark Selection. Our results further justify selecting VINE, TrustMark, and VideoSeal as the building blocks of TAILOR. The three watermarks exhibit complementary robustness: TrustMark is strongest on S1 (100%), VideoSeal on S2 (49.74%), and VINE provides stronger coverage on S3 and S4 (42.68% and 54.26%). Their distinct strengths motivate the fragment library. We further evaluate whether these fragments remain compatible when composed. In the direct superposition experiment (Appendix D.3), their mean raw bit accuracy drops by only 0.49 percentage points after composition, and all fragment readouts remain above the verification threshold under suitable or-

Table 2: Ablation of candidate search, composition, and strength adaptation.

	Satisfaction	*Image fidelity*		
Method	Rate (%) ↑	PSNR ↑	SSIM ↑	PSNR-P
Single-fragment	63.64	**41.84**	0.9919	41.84 / 42.20
Unit-strength	73.18	37.85	0.9887	37.92 / 41.40
SMT-top1	80.65	<u>41.22</u>	**0.9927**	41.22 / 41.22
SMT-top3	<u>91.99</u>	40.83	<u>0.9920</u>	40.83 / 40.83
TAILOR-F	**95.50**	40.66	0.9918	-

PSNR-P: row variant / TAILOR-F, evaluated on jointly accepted requests.

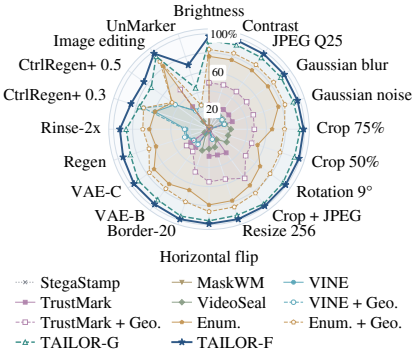


Figure 5: Request satisfaction grouped by attack inclusion.

ders. The recovery-interaction ablation (Appendix D.4) further changes request satisfaction by only 0.09 percentage points when explicit interaction corrections are removed. Together, these results show that the selected watermarks provide complementary robustness while remaining sufficiently compatible for joint composition, supporting their use as the fragment library of TAILOR.

5.3 Ablation Studies

Candidate Search and Live Refinement. We evaluate whether broader candidate search and live feedback improve request satisfaction. As shown in Table 2, SMT-top1 satisfies 80.65% of requests, while evaluating up to three candidate structures increases satisfaction to 91.99%, a gain of 11.34 percentage points. Request-local refinement and margin fallback further raise satisfaction to 95.50%, adding another 3.51 points. This shows that the highest-ranked offline candidate is not always sufficient and that broader search and live refinement recover additional feasible deployments. On jointly accepted requests, SMT-top1 and SMT-top3 select the same configurations as TAILOR, yielding identical mean PSNRs of 41.22 and 40.83 dB on their respective intersections. Thus, the additional search expands coverage without degrading fidelity on already successful requests.

Watermark Composition. We evaluate whether optimizing a single watermark per request can replace composition. Single-fragment retains strength optimization, geometric recovery stage selection, candidate search, and the full live-validation procedure, but satisfies only 63.64% of requests, compared with 95.50% for TAILOR. The 31.86-percentage-point gap shows that a single optimized watermark is insufficient for requests involving heterogeneous attacks. On jointly accepted requests, TAILOR achieves 42.20 dB compared with 41.84 dB for Single-fragment, showing that composition improves both coverage and fidelity.

Strength Adaptation. We evaluate the contribution of per-request strength optimization by fixing all selected watermark strengths to one. Unit-strength retains subset, order, geometric recovery stage, candidate search, and live refinement, but satisfies only 73.18% of requests, compared with 95.50% for TAILOR. On jointly accepted requests, TAILOR achieves 41.40 dB versus 37.92 dB for Unit-strength, a 3.48 dB gain. These results show that strength adaptation is important for jointly improving robustness and image fidelity. Additional ablation studies are provided in Appendix D.

6 Conclusion

In this paper, we propose TAILOR, a framework for composing image watermarks to meet deployment-specific requirements. TAILOR combines SMT-based joint configuration selection with live calibration to choose complementary watermark fragments and embedding settings under specified attacks, FPR budgets, quality floors, and latency budgets. Experiments on 7,321 distinct requests across five scenarios demonstrate higher request satisfaction than fixed single-watermark and shared-model greedy baselines, together with improved image quality on jointly accepted requests. This work provides a basis for adapting watermark composition to the provenance and attribution needs of different applications.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data sets, to help develop theoretical models and conceptual frameworks, to propose and refine hypotheses, to design and provide feedback on research methodology and experiments, to implement methods, to clean and reformat datasets, and to interpret results. We have not used generative AI tools to assist with translation or to support qualitative and thematic data analysis, and formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work, which states no formal claim with a proof. Additionally, we used generative AI tools to create and edit software code, to create and modify scientific figures, to suggest experimental parameters, to edit it for readability and structure, to propose a title, to brainstorm, and to search for information.

We have reviewed all AI-assisted work. Every number reported here is emitted by code from the frozen measurement records rather than transcribed by hand, and each table and figure is re-derived from one committed result file, so a reported value traces to the measurement that produced it. The protocol enforces by file, byte and pixel digests that the slice used to fit the offline model and the slice every reported number is measured on share no image. AI-assisted code was verified by the authors through independent recomputation of the reported quantities and through the unit tests released with the code. AI-drafted text and AI-suggested claims were checked against the measurements, and claims the measurements did not support were removed. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Adobe. Impact of applying Content Credentials. Adobe Help Center, September 2025. URL https://helpx.adobe.com/creative-cloud/apps/adobe-content-authenticity/impact.html. Updated September 2, 2025. Accessed September 18, 2026.

Eirikur Agustsson and Radu Timofte. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

Ali Al-Haj. Combined DWT-DCT digital image watermarking. *Journal of computer science*, 3(9): 740–746, 2007.

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

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

Clark Barrett and Cesare Tinelli. Satisfiability modulo theories. In *Handbook of model checking*, pp. 305–343. Springer, 2018.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402. IEEE, 2023.

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

Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7936–7945. IEEE, 2020.

Coalition for Content Provenance and Authenticity. C2PA Technical Specification. https://spec.c2pa.org/, 2024.

Leonardo de Moura and Nikolaj Bjørner. Z3: An Efficient SMT Solver. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 2008.

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

Pierre Fernandez, Tomáš Souček, Nikola Jovanović, Hady Elsahar, Sylvestre-Alvise Rebuffi, Valeriu Lacatusu, Tuan Tran, and Alexandre Mourachko. Geometric image synchronization with deep watermarking. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 9522–9526. IEEE, 2026.

Google DeepMind. Nano banana 2: Combining pro capabilities with lightning-fast speed. https://blog.google/innovation-and-ai/technology/ai/nano-banana-2/, February 2026. Accessed: 2026-09-25.

Sven Gowal and Pushmeet Kohli. Identifying AI-generated images with SynthID. Google DeepMind Blog, August 2023. URL https://deepmind.google/blog/identifying-ai-generated-images-with-synthid/. Published August 29, 2023. Accessed September 18, 2026.

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

Runyi Hu, Jie Zhang, Shiqian Zhao, Nils Lukas, Jiwei Li, Qing Guo, Han Qiu, and Tianwei Zhang. Mask image watermarking. *Advances in Neural Information Processing Systems*, 38:146313–146346, 2026.

Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):1, 2008.

Zhengyuan Jiang, Jinghuai Zhang, and Neil Zhenqiang Gong. Evading watermark based detection of ai-generated content. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1168–1181, 2023.

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

Yawei Li, Kai Zhang, Jingyun Liang, Jiezhang Cao, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu, Denis Demandolx, et al. LSDIR: A large scale dataset for image restoration. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1775–1787. IEEE, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

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

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

Meta. What's new across our ai experiences. https://about.fb.com/news/2023/12/meta-ai-updates/, December 2023. Accessed: 2026-09-25.

KA Navas, Mathews Cheriyan Ajay, M Lekshmi, Tampy S Archana, and M Sasikumar. Dwt-dct-svd based watermarking. In *2008 3rd international conference on communication systems software and middleware and workshops (COMSWARE'08)*, pp. 271–274. IEEE, 2008.

Natalia Ellen Nicholas, Furqan Rustam, and Anca Delia Jurcut. Flexmark: Evaluating watermark robustness and attribution reliability under redistribution and adaptive ai laundering. In *2026 IEEE Symposium on Security and Privacy Workshops (SPW)*, pp. 238–249. IEEE, 2026.

OpenAI. Gpt-image-2 model. https://developers.openai.com/api/docs/models/gpt-image-2, 2026. Accessed: 2026-09-25.

Andy Parsons. Introducing Adobe Content Authenticity: A free web app to help creators protect their work, gain attribution and build trust. Adobe Blog, October 2024. URL https://blog.adobe.com/en/publish/2024/10/08/introducing-adobe-content-authenticity-free-web-app-help-creators-protect-their-work-gain-attribution-build-trust. Published October 8, 2024. Accessed September 18, 2026.

Aleksandar Petrov, Shruti Agarwal, Philip Torr, Adel Bibi, and John Collomosse. On the coexistence and ensembling of watermarks. *Advances in Neural Information Processing Systems*, 38:107058–107094, 2026.

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*, volume 2024, pp. 1862–1874, 2024.

Tom Sander, Pierre Fernandez, Alain Oliviero Durmus, Teddy Furon, and Matthijs Douze. Watermark anything with localized messages. In *International Conference on Learning Representations*, volume 2025, pp. 79569–79599, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/c5ee1911dffe4886367eee7fca314235-Paper-Conference.pdf.

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

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

Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 893–911, 2023.

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

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3008–3018. IEEE, 2025.

Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024a.

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

A IMPLEMENTATION DETAILS

This appendix provides the implementation details for the three stages of TAILOR, including the performance database, SMT encoding, FPR accounting, live calibration, and watermark deployment. For reference, Table 3 summarizes the notation used throughout the formulation and the main text.

A.1 PRELIMINARIES AND NOTATION

Piecewise-Linear Response Curves. The offline database measures recovery, distortion, and latency only at a finite set of embedding strengths, while configuration selection optimizes strength continuously. We therefore represent each measured quantity using the piecewise-linear interpolation in Eq. 1. For consecutive knots $\lambda^{(0)} < \cdots < \lambda^{(K)}$, the SMT encoding introduces one Boolean variable z_k for each interval and enforces exactly one active segment:

$$\bigvee_{k=0}^{K-1} z_k, \qquad \neg(z_k \wedge z_{k'}) \quad \forall\, k \neq k', \qquad z_k \Rightarrow \Big[\lambda^{(k)} \leq \lambda \leq \lambda^{(k+1)} \wedge \widehat{y} = y^{(k)} + s^{(k)}\big(\lambda - \lambda^{(k)}\big)\Big]. \tag{10}$$

The knots $\lambda^{(k)}$, measured values $y^{(k)}$, and slopes $s^{(k)}$ are fixed constants, so every active segment contributes only linear constraints over the strength variable. Because no segment is defined outside $[\lambda^{(0)}, \lambda^{(K)}]$, the solver may interpolate between measured strengths but never extrapolates beyond the characterized range.

Recovery curves $b_f(\cdot, a)$ and interaction curves $\delta_{g\to f}(\cdot, a)$ are maintained separately for each attack. Standalone distortion $D_f(\cdot)$ and latency are modeled per fragment, while the pairwise distortion correction e_{fg} is indexed by $\lambda_f + \lambda_g$.

SMT Formulation. Configuration selection couples discrete structural choices with continuous embedding strengths. Fragment selection, embedding order, and geometric recovery are discrete decisions, whereas each selected fragment has a continuous native strength; these variables interact because predicted recovery and distortion depend on the complete composition. SMT (Barrett & Tinelli, 2018) is therefore well suited to the problem, since it combines Boolean structure with arithmetic constraints in a single model.

We encode all request requirements and configuration-validity conditions as an SMT formula over Boolean variables and linear real arithmetic. A satisfying assignment directly specifies a feasible configuration, while an UNSAT result indicates that no configuration satisfies the encoded requirements within the characterized search space. We use Z3 (de Moura & Bjørner, 2008) to solve the resulting model.

Optimization is performed by repeated satisfiability queries. After obtaining a feasible configuration with objective value D_{best}, we add the constraint that the next solution must improve distortion by at least ϵ and solve again. This process continues until no improving assignment exists, yielding the lowest-distortion configuration up to the specified tolerance.

A.2 PERFORMANCE DATABASE

This section gives the implementation details of the offline quantities introduced in Sec. 4.1, including how recovery interactions, distortion corrections, geometric responses, and runtime are represented in the Performance Database. Each strength-dependent quantity is stored as the PWL response described in Sec. A.1, together with the per-image measurements needed by request-local calibration.

Recovery. For each fragment–attack pair, the database stores the standalone response $b_f(\lambda_f, a)$ and the corresponding per-image scores. Ordered composition effects are represented by $\delta_{g\to f}(\lambda_g, a)$, measured as the change in f's recovery when g is embedded after it. Thus, for a configuration with order π, Eq. 2 subtracts the measured correction for every g satisfying $f \prec_\pi g$. Supported geometric recovery stages maintain separate response curves for the affected fragment–attack pairs.

Image Quality. The database stores standalone embedding distortion $D_f(\lambda_f)$ in MSE and a pairwise correction e_{fg} for deviations from additive distortion under sequential embedding. To keep the correction univariate, e_{fg} is indexed by $\lambda_f + \lambda_g$. Together with the geometric-stage term $D_\varphi(c)$, these quantities instantiate Eq. 3.

Table 3: Core notation used throughout the method and appendix.

Symbol	Meaning
Deployment request	
$u = (\mathcal{A}_u, \alpha, q, t)$	Deployment request specifying the attack set, FPR budget, PSNR floor, and latency ceiling
$\mathcal{A}_u$	Set of requested attacks; a denotes one attack
α	Configuration-level false-positive-rate budget for one image verification
$q,\ t$	Minimum PSNR in dB and maximum deployment latency in ms
Configuration	
$f,\ g$	Watermark fragments drawn from VINE, TrustMark, and VideoSeal
$c = (S, \lambda, \pi, \varphi)$	Configuration specifying the selected fragments, native strengths, embedding order, and geometric recovery stage
$S,\ m = \lvert S \rvert$	Selected fragment subset and number of selected fragments
λ_f	Native embedding strength of fragment f within its characterized range
$\pi,\ f \prec_\pi g$	Embedding order; $f \prec_\pi g$ means that g is embedded after f
φ	Enabled geometric recovery stage, at most one of scale, tile, resync, and angle
$E_c,\ D_c$	Composite encoder and configured decoder induced by c
Performance database	
$b_f(\lambda_f, a)$	Measured standalone mean bit accuracy of fragment f at strength λ_f under attack a
$\delta_{g \to f}(\lambda_g, a)$	Measured recovery loss of f caused by embedding g after it
$\widehat{b}_f(c, a)$	Predicted mean bit accuracy of fragment f under configuration c and attack a
$\Gamma_{f,a}(c)$	Whether f is predicted to satisfy the screening acceptance-rate requirement $\bar{\rho}$ under attack a
$D_f(\lambda_f)$	Measured standalone embedding MSE of fragment f
$e_{fg}(\lambda_f + \lambda_g)$	Measured pairwise excess MSE beyond the sum of standalone distortions
$D_\varphi(c)$	Additional embedding MSE introduced by the enabled geometric recovery stage
$D(c),\ P(c)$	Predicted total embedding MSE and corresponding PSNR
$L(c)$	Predicted deployment latency of configuration c in ms
Verification	
$w,\ \ell_j$	Expected 100-bit codeword and aligned log-likelihood ratio for bit j
$B(\ell, w)$	Bit-match accuracy of one aligned readout against w
$T(c)$	Maximum number of verification tests evaluated by configuration c
$\theta(c, \alpha),\ \tau(c, \alpha)$	Integer match threshold and its normalized value in $[0, 1]$
ρ	Required fraction of images that must pass verification under each requested attack
$\eta,\ \eta_{\text{rate}}$	Screening margins applied to the recovery threshold and acceptance rate during selection
Live calibration	
$s_i(c, a),\ N_a$	Verification score for image i under configuration c and attack a, and the number of evaluated images
$\bar{b}$, se	Live mean recovery and its standard error
Δb	Request-local downward correction applied to an optimistic recovery prediction
c_{out}	Configuration accepted after live calibration for deployment

The pairwise correction is small for the deployed library: it is zero over the measured range for TrustMark–VideoSeal and VINE–VideoSeal, and at most 0.35 MSE for TrustMark–VINE, compared with standalone values of 1–21 MSE. Across the three-fragment orders, the resulting distortion model differs from measured PSNR by at most 0.09 dB at mixed strengths and 0.31 dB near the maximum strengths.

Runtime. The latency model sums the measured embedding and decoding costs of the selected fragments and adds the overhead of the enabled geometric recovery stage. These measurements instantiate the runtime term $L(c)$ used by the solver.

A.3 Decoder and FPR Accounting

A configuration may expose multiple verification paths through individual fragments, fused readouts, and geometric recovery views. Evaluating more paths can improve watermark recovery but also increases the opportunity for false positives. TAILOR therefore uses a common verification score for all readouts and accounts for every permitted test when setting the detection threshold.

Unified Verification Score. Let $w \in \{0,1\}^{100}$ denote the expected watermark codeword and ℓ_j the aligned log-likelihood ratio for bit position j. Every readout is scored by its bit accuracy against w:

$$B(\ell, w) = \frac{1}{100} \sum_{j=1}^{100} \mathbf{1}[(2w_j - 1)\ell_j > 0]\,. \tag{11}$$

Each fragment's key-derived permutation and sign mask are inverted before scoring. When multiple fragments are selected, their aligned log-likelihoods may additionally be fused and evaluated using the same rule. Individual, fused, and geometric readouts therefore share a common verification statistic.

Verification-Test Budget. Let $T(c)$ denote the maximum number of verification tests permitted by configuration c:

$$\begin{aligned} T(c) &= T_0(m) + 5mz_r + 133z_s + 81z_d + 65mz_a, \\ T_0(m) &= \begin{cases} 1, & m = 1, \\ m + 1, & m > 1, \end{cases} \end{aligned} \tag{12}$$

where $m = |S|$, and $z_r, z_s, z_d, z_a \in \{0,1\}$ indicate whether resynchronization, scale search, tile search, and angle search are enabled, respectively. The base term $T_0(m)$ counts the individual fragment readouts and, when $m > 1$, one additional fused readout. The remaining terms count the candidate views introduced by the corresponding geometric recovery stages. Thus, the complete verification-test budget is determined by the selected configuration before an image is evaluated.

Null Model. We calibrate the verification threshold under the null hypothesis that the evaluated image does not contain the deployed watermark. Let $\Pr_0$ denote probability under this null, averaging over secret-key generation and negative images for a fixed configuration and watermark codeword.

For each permitted readout r, let J_r denote the bit positions with nonzero aligned log-likelihoods. We assume that, under the null, the signs at these positions are mutually independent and equally likely to agree or disagree with the expected codeword. If $K_r = 100B_r$ denotes the number of matching bits, then

$$K_r \mid J_r \sim \mathrm{Bin}(|J_r|, 1/2), \qquad \Pr_0[K_r \geq h] \leq \Pr[X \geq h], \quad X \sim \mathrm{Bin}(100, 1/2). \tag{13}$$

Zero log-likelihoods are counted as mismatches, so the 100-bit binomial tail remains an upper bound when some bit positions are unresolved. This assumption concerns the aligned decoder outputs with the expected codeword held fixed and does not require independence among the bits of the codeword itself.

Configuration-Level FPR Control. Because configuration c may evaluate up to $T(c)$ verification tests, we allocate each test a false-positive budget of $\alpha/T(c)$. We then choose the smallest number of matching bits whose binomial upper-tail probability satisfies this budget:

$$\theta(c, \alpha) = \min\left\{ h \in \{0, \ldots, 101\} : \Pr[X \geq h] \leq \frac{\alpha}{T(c)} \right\}, \qquad \tau(c, \alpha) = \frac{\theta(c, \alpha)}{100}. \tag{14}$$

The sentinel value $h = 101$ represents an impossible-pass threshold when no attainable 100-bit match count satisfies the requested per-test budget. A verification test passes only when its score reaches $\tau(c, \alpha)$.

Under the null model, every permitted test therefore has false-positive probability at most $\alpha/T(c)$. Applying a union bound over all permitted tests gives

$$\Pr_0(\text{accept}) \le \sum_{r=1}^{T(c)} \Pr_0[B_r \ge \tau(c,\alpha)] \le T(c)\frac{\alpha}{T(c)} = \alpha. \tag{15}$$

Thus, configurations with more verification paths automatically receive a stricter threshold, while the overall per-image false-positive probability remains bounded by the requested budget α under the null model.

A.4 Per-Image Acceptance

Offline selection uses both predicted mean recovery and stored image-level acceptance rates. At each strength, the acceptance rate is the fraction of stored fragment scores reaching $\tau(c, \alpha)$. A nondecreasing isotonic fit maps this rate to the minimum strength satisfying $\bar{\rho} = \min(1, \rho + \eta_{\text{rate}})$ and defines $\Gamma_{f,a}(c)$. We use $\rho = 0.90$, initial rate margins $\eta_{\text{rate}} \in \{0.04, 0.06\}$, and initial mean-accuracy margin $\eta = 0.02$. Clean-image sweeps additionally impose minimum strengths where needed.

A.5 Request-Local Refinement

Offline predictions may not transfer accurately to the user's images. Live calibration therefore uses failed candidate evaluations to identify where the performance database is optimistic, applies conservative request-specific corrections, and re-solves the configuration problem. The correction is local to the current request and affects only predictions supported by the observed live measurements.

Live Discrepancy Measurement. Candidates are evaluated in predicted-quality order, with up to three distinct discrete structures and optimized continuous strengths. For each tested candidate, requested attack a, and selected fragment f, live validation produces per-image recovery scores at the strength λ_f used by that candidate. We compare two live statistics with the corresponding offline predictions: the mean recovery $\bar{b}$, with standard error $\text{se} = \hat{\sigma}/\sqrt{N_a}$, and the image-level acceptance rate, i.e., the fraction of images whose configured readout reaches the verification threshold $\tau(c, \alpha)$. These discrepancies provide request-specific evidence about where the offline model overestimates recovery.

Mean-Recovery Correction. Let $\widehat{b}$ be the recovery predicted by the database at the same configuration and strength, and define the observed gap $g = \bar{b} - \widehat{b}$. We do not directly replace the offline prediction with the live mean, since the live measurement itself is estimated from a finite image set. Instead, we apply the shrinkage correction

$$\Delta b = \begin{cases} 0, & |g| < 2\,\text{se}, \\ \dfrac{\sigma_0^2}{\sigma_0^2 + \text{se}^2}\, g, & \text{otherwise}, \end{cases} \tag{16}$$

and retain only corrections that reduce the predicted recovery. The $2\,\text{se}$ gate ignores discrepancies consistent with sampling variation, while the shrinkage factor gives greater weight to precise live measurements and less to uncertain ones. We set $\sigma_0 = 0.023$, the 90th percentile of the measured variation of cell means across image sources using 30 images per source. Restricting corrections to the pessimistic direction prevents favorable sampling noise from creating an unsupported SAT configuration.

Acceptance-Rate Correction. The image-level acceptance prediction is derived from stored per-image scores, so a discrepancy in acceptance rate cannot be added directly as a scalar offset. Instead, we shift the stored score distribution downward until its predicted acceptance rate matches the observed live rate, solving for the shift by bisection over $[-0.4, 0]$. If the observed rate falls below the required ρ, the corresponding downward correction is always applied. Otherwise, we apply a correction only when the predicted rate exceeds the observed rate by more than two standard errors. When mean-recovery and acceptance-rate evidence produce corrections to the same cell, we retain the more conservative one.

Request-Local Model Update. Each accepted correction shifts the corresponding response curve rather than only the measured strength. A pointwise update could allow the next solve to move to a neighboring strength on the same optimistic curve and reproduce the failed configuration. Shifting the curve preserves its measured shape while correcting its request-specific level; the associated per-image scores are shifted consistently so that mean and acceptance-rate predictions remain aligned. If the failed candidate used a geometric recovery stage, the correction is applied to that stage's response curve, since that is the prediction used by the solver.

All corrections are applied to a request-local copy of the Performance Database and never modify the shared offline measurements. TAILOR then re-solves the joint configuration problem once using the corrected model, and the resulting candidate must pass the same live validation criteria before it can be accepted.

A.6 WATERMARK DEPLOYMENT

The composition library contains VINE (Lu et al., 2025), TrustMark (Bui et al., 2023), and VideoSeal (Fernandez et al., 2024). Their native controls are VINE residual scale, TrustMark embedding strength, and VideoSeal blend scale. Every fragment transmits the same 100-bit word: VINE and TrustMark carry 100 bits natively, whereas VideoSeal's message vector is longer, so the deployment writes the word into its first 100 positions and reads back the corresponding 100 logits, leaving the rest unused. Sequential embedding in the accepted order follows Eq. 4, with w entering each fragment through that fragment's own key-derived permutation and mask.

At verification, the decoder reverses each fragment's key-derived permutation and mask, evaluates the permitted individual readouts and primary fusion, and then applies the enabled geometric search when needed. All paths use the same configuration-dependent threshold in Eq. 14.

B GEOMETRIC RECOVERY STAGES

Geometry is the attack family where the three fragments fail for different reasons, so we first measure where each fragment writes its payload and then derive one recovery stage per failure mode. Figure 6 shows the resulting stages: the residual each one writes into the image, the attack it answers, and the views it enumerates at decoding; its residuals use a $12\times$ display gain. The paragraphs below give the mechanism and the parameters.

B.1 WHERE EACH FRAGMENT WRITES ITS PAYLOAD

We revert part of a watermarked image to the cover and read the fragment from the result. Reverting the interior leaves only a border ring, the outer tenth of the half-width; reverting that ring leaves everything else.

Table 4: Where each fragment writes its payload, measured by reverting part of the watermarked image to the cover.

	Bit accuracy after reverting			*Residual energy*
Fragment	Nothing	Interior	Border ring	in the border ring (%)
VINE	1.000	**1.000**	0.482	**50.7**
TrustMark	1.000	0.500	**1.000**	7.1
VideoSeal	0.997	0.652	0.985	15.7

The border ring is the outer tenth of the half-width, that is 19% of the area. Means over 24 images at unit strength.

Table 4 reports the result, and VINE and TrustMark come out opposite. VINE reads perfectly from the border ring alone and at chance from the interior alone, and half of its residual energy sits in that ring, which is a fifth of the area: its payload is a thin frame at one radius. TrustMark is the reverse. It reads perfectly from the interior and at chance from the ring, spreading the codeword across the frame with a centre bias. VideoSeal is close to uniform.

Their behaviour under geometry follows from this. On the frozen measurements at $\alpha = 10^{-4}$, VINE alone accepts no centred crop, no rotation, and no flip, while still accepting every image under JPEG: it is a single fixed-coordinate template that any geometric change moves off its grid.

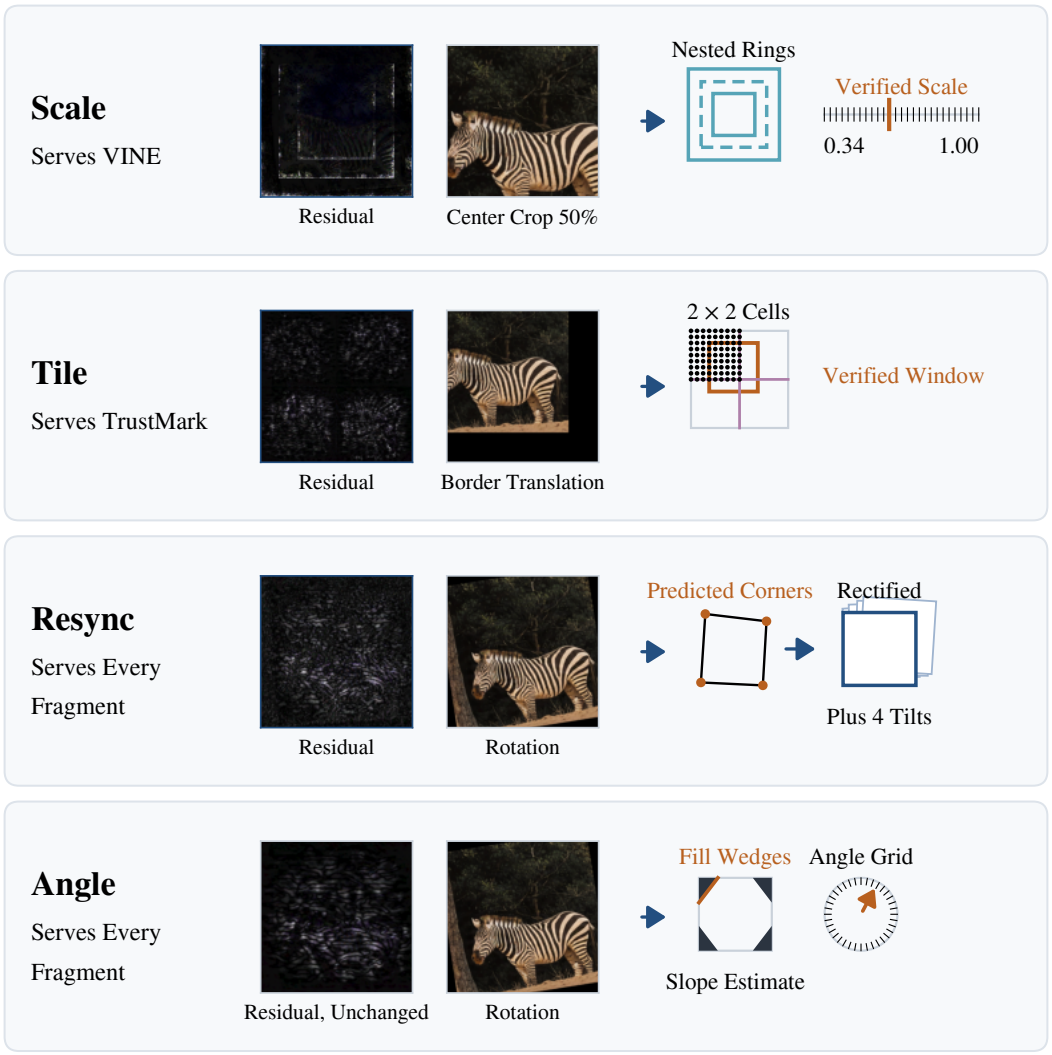


Figure 6: The four optional geometric stages, with the embedded residual each one produces and the views it searches at decoding.

TrustMark alone accepts a crop to 75% of the area, because enough of a spread codeword remains, but drops to 1% once a border translation shifts the frame and to 4% under a 9 degree rotation.

B.2 Failure-Driven Geometric Recovery

Scale, for a payload on one radius. A ring occupies a single radius, so a centred crop of side factor s moves a ring embedded at radius K to K/s and cuts off every ring with $K > s$. Embedding the same codeword at several radii therefore leaves one ring inside the frame for any crop down to the smallest radius, and recovery reduces to finding which one: the decoder reads VINE at a dense grid of centred scales and keeps the first view that verifies. We embed at $K \in \{1.00, 0.75, 0.50\}$. Each radius covers a different crop depth. Under a crop to half the area only the innermost ring is still inside the frame, and it is read at scale $0.50/\sqrt{0.5} \approx 0.71$; the outer two are gone. The smallest radius is 0.50 because VINE encodes at 256×256 on a 512×512 canvas, so that ring is the smallest one that needs no resampling. The decoder reads VINE at 133 centred scales from 0.34 to 1.00 in steps of 0.005. The step is 0.005 because a ring only reads within roughly 1% of its own scale, which leaves no gradient for a coarse-to-fine search to follow, and the range starts at 0.34 because no smaller view can hold the innermost ring.

Tile, for a payload on one grid. A spread codeword survives a centred crop that keeps enough area, but it is still written on one fixed grid, so a translation moves every position off that grid at once. Repeating the whole codeword in each cell of a 2×2 grid removes the dependence on where the frame starts: any window of one cell size that lands on a cell carries a complete codeword. The decoder slides a 256 px window over the frame at a stride of 32 px, 81 positions in all, and verifies each one, so recovery needs a surviving cell rather than a surviving frame.

Resync, for a warped frame. A rotation or a homography does not remove the payload; it moves it away from where every decoder looks, which is why all three fragments fail together under it. One estimate of the frame therefore repairs all of them at once, and it is worth spending embedding capacity on making that estimate possible: the stage appends a SyncSeal template (Fernandez et al., 2026) whose predicted corners give the homography back to the original frame. The decoder rectifies once and reads every selected fragment from the result, then repeats at tilts of 3 and 6 degrees either way to absorb the residual error of the corner prediction, five views per fragment in all.

Angle, for rotation without a template. Rotation leaves evidence in the image itself: an in-place rotation pads the corners with fill, and the boundary between fill and content is a straight line whose slope is the rotation. Reading the angle from the frame rather than from a fragment costs nothing at embedding and keeps the estimate independent of which fragments a request selected, so a rotation-fragile choice of fragments cannot disable it. The decoder proposes the angles implied by the corner wedges first, then falls back to a fixed sweep on a 10 degree grid so that the search stays complete when no wedges are present, at most 65 views per fragment.

Scope. Scale and tile change how one fragment is written, so each applies only to that fragment and only that fragment is read from their views. Resync and angle restore the frame, so every selected fragment is read from each of their views. This is the origin of the per-stage test counts in Eq. 12: the two frame-restoring stages scale with the number of selected fragments, the two fragment-specific stages do not.

C Evaluation Details

C.1 Image Pool and Measurement Roles

Images are drawn from five source groups:

- **MS-COCO 2017 (Lin et al., 2014):** real photographs with diverse native resolutions.
- **DiffusionDB (Wang et al., 2023):** SD-1.5 generated images at 512×512.
- **UltraEdit (Zhao et al., 2024a):** real images and image-editing examples at 512×512.
- **DALL·E 3 / SDXL (Podell et al., 2024):** generated images at 1024×1024.
- **DIV2K (Agustsson & Timofte, 2017) and LSDIR (Li et al., 2023):** high-resolution real images ranging from 1024 pixels to approximately 2K resolution.

All experiments use three fixed, mutually disjoint slices of 100 images, each containing 25, 25, 25, 15, and 10 images from the five source groups above. Two slices are used exclusively for offline

characterization: one for fragment strength sweeps, pairwise interaction measurements, distortion modeling, and geometric recovery stages, and the other for regeneration, rinse, CtrlRegen+, and watermark-removal attacks. The third slice serves as the live set for request-local calibration and all reported evaluations, and is shared across requests. File-, byte-, and pixel-level digest checks confirm that the three slices contain no overlapping images, ensuring strict separation between offline characterization and live evaluation.

C.2 Request Distribution and Attack Settings

We construct deployment requests over attacks with different adversarial capabilities. In all settings, the adversary has access to the released watermarked image x_w, but not to the protector's secret key or the original image x_o. We group additional capabilities into three levels: *Image Manipulation*, covering standard image transformations; *Public-Model Access*, which additionally permits the use of publicly available pretrained models; and *Optimization-Based Removal*, which allows iterative optimization directly on the released image.

The watermark-removal scenario (S4) includes CtrlRegen+ at strength 0.3 or 0.5, optionally combined with UnMarker. Both target watermark removal but use different mechanisms: CtrlRegen+ performs controlled regeneration, whereas UnMarker applies iterative optimization.

Request Composition. The main benchmark contains 7,321 distinct deployment requests spanning 20 attack settings and 240 unique attack sets. S4 contributes 822 requests, including 548 using CtrlRegen+ at strength 0.3 and 274 at strength 0.5; 262 of these requests additionally include UnMarker. All compared methods are evaluated on the same request IDs. Table 5 summarizes the request distribution across scenarios and FPR budgets, with the corresponding quality and latency budgets specified in the table note.

Table 5: The 7,321 distinct request specifications: what each scenario contributes, and how its requests divide over the six FPR budgets.

		Distinct specifications			*Specifications per FPR budget*					
ID	Scenario	Count	Share (%)	Attack sets	10^{-1}	10^{-2}	10^{-4}	10^{-6}	10^{-9}	2^{-37}
S1	Signal / re-encoding	1,551	21.19	32	312	441	345	221	120	112
S2	Geometry	1,532	20.93	32	309	434	355	215	114	105
S3	Regeneration / editing	1,373	18.75	16	277	328	313	223	125	107
S4	Watermark removal	822	11.23	32	362	188	155	91	9	17
S5	Broad attack coverage	2,043	27.91	128	438	628	489	274	111	103
Total		7,321	100.00	240	1,698	2,019	1,657	1,024	479	444

PSNR floors in dB are drawn from $\{34, 35, 36, 37, 38, 40\}$ in S1, $\{32, 33, 34, 35, 36, 38\}$ in S2 and S5, and $\{30, 31, 32, 33, 34, 36\}$ in S3 and S4; latency budgets in ms from $\{500, 1000, 2000, 4000\}$ in S1, $\{1000, 2000, 4000, 8000\}$ in S2 and S3, $\{2000, 4000, 8000\}$ in S4, and $\{4000, 8000, 16000\}$ in S5.

C.3 Baseline

The main comparison includes the full method, the greedy variant, fixed single-watermark controls, optional-geometry single-watermark controls, Enumeration with and without geometry, MaskWM, and StegaStamp.

Greedy. The greedy variant TAILOR-G starts from the empty fragment subset and irrevocably adds the fragment with the largest predicted coverage gain. For each trial subset, the full method's SMT model reoptimizes continuous strengths, embedding order, and compatible geometry. The first complete candidate satisfying the request is passed to the common live-validation procedure.

Fixed Single-Watermark Controls. VINE, TrustMark, and VideoSeal are evaluated at unit strength without geometry. Their optional-geometry variants additionally try compatible geometric stages using the same implementations and FPR accounting as TAILOR.

MaskWM and StegaStamp. MaskWM (Hu et al., 2026) and StegaStamp (Tancik et al., 2020) are evaluated as fixed external baselines. Both embed the same 100-bit transmitted word and use the common strict acceptance rule. StegaStamp's evaluated resize pipeline has mean embedding

Table 6: The 20 attack settings used in evaluation, with the exact configuration each one runs.

Attack	Capability	Configuration
Photometric and signal processing		
Brightness	Image Manipulation	factor 0.2
Contrast	Image Manipulation	factor 0.2
JPEG Q25	Image Manipulation	quality 25
Gaussian blur	Image Manipulation	kernel 5, $\sigma = 1$
Gaussian noise	Image Manipulation	$\sigma = 0.05$
Geometric transformations		
Crop 75%	Image Manipulation	centre crop keeping 75% of the area, side factor 0.866; canvas restored
Crop 50%	Image Manipulation	centre crop keeping 50% of the area, side factor 0.707; canvas restored
Rotation 9°	Image Manipulation	9° in place, corners left as fill
Crop + JPEG	Image Manipulation	centre crop to 75% area, then JPEG quality 25
Resize 256	Image Manipulation	downscale to 256, resize back to 512
Horizontal flip	Image Manipulation	horizontal mirroring
Border-20	Image Manipulation	translation crop: drop 20% of the side from the top and left, push the remainder to the origin, leave the vacated strips empty; frame size preserved
Neural compression		
VAE-B (Ballé et al., 2018)	Public-Model Access	`bmshj2018-hyperprior`, quality level 3
VAE-C (Cheng et al., 2020)	Public-Model Access	`cheng2020-anchor`, quality level 3
Regeneration and editing		
Regen (Zhao et al., 2024b)	Public-Model Access	SD 2-1, DPMSolver, `noise_step`$= 60$, one pass
Rinse-2x	Public-Model Access	two successive Regen passes
CtrlRegen+ 0.3 (Liu et al., 2025)	Public-Model Access	strength 0.3
CtrlRegen+ 0.5 (Liu et al., 2025)	Public-Model Access	strength 0.5
Image editing (Brooks et al., 2023)	Public-Model Access	InstructPix2Pix, guidance 7.0, image guidance 1.5, five global prompts
Watermark removal		
UnMarker (Kassis & Hengartner, 2025)	Optimization-Based Removal	spectral and perceptual optimization on the released image

Every setting is a single fixed configuration, not a range: a request either contains the setting or it does not. The adversary sees the released watermarked image only.

Table 7: Request satisfaction with 95% confidence intervals. Each entry is reported as $p^{+\Delta_{\text{upper}}}_{-\Delta_{\text{lower}}}$ in percentage points.

	Request satisfaction (%) ↑ with 95% CI					
Method	S1	S2	S3	S4	S5	**Avg.**
StegaStamp	$0.00^{+0.25}_{-0.00}$	$0.00^{+0.25}_{-0.00}$	$0.00^{+0.28}_{-0.00}$	$0.00^{+0.47}_{-0.00}$	$0.00^{+0.19}_{-0.00}$	$0.00^{+0.13}_{-0.00}$
MaskWM	$10.32^{+1.61}_{-1.42}$	$0.00^{+0.25}_{-0.00}$	$0.00^{+0.28}_{-0.00}$	$0.00^{+0.47}_{-0.00}$	$0.00^{+0.19}_{-0.00}$	$2.06^{+0.35}_{-0.28}$
VINE	$8.83^{+1.52}_{-1.31}$	$0.00^{+0.25}_{-0.00}$	$42.68^{+2.63}_{-2.59}$	$54.26^{+3.38}_{-3.42}$	$0.00^{+0.19}_{-0.00}$	$21.15^{+0.91}_{-0.90}$
TrustMark	$100.00^{+0.00}_{-0.25}$	$0.00^{+0.25}_{-0.00}$	$23.38^{+2.31}_{-2.16}$	$0.00^{+0.47}_{-0.00}$	$0.00^{+0.19}_{-0.00}$	$24.68^{+0.48}_{-0.44}$
VideoSeal	$27.98^{+2.29}_{-2.18}$	$49.74^{+2.50}_{-2.50}$	$4.01^{+1.17}_{-0.92}$	$0.00^{+0.47}_{-0.00}$	$2.74^{+0.80}_{-0.62}$	$16.89^{+0.74}_{-0.70}$
VINE + Geo.	$8.83^{+1.52}_{-1.31}$	$0.00^{+0.25}_{-0.00}$	$42.68^{+2.63}_{-2.59}$	$62.65^{+3.24}_{-3.36}$	$0.00^{+0.19}_{-0.00}$	$22.83^{+0.89}_{-0.89}$
TrustMark + Geo.	$100.00^{+0.00}_{-0.25}$	$94.71^{+1.01}_{-1.24}$	$23.38^{+2.31}_{-2.16}$	$0.00^{+0.47}_{-0.00}$	$11.94^{+1.48}_{-1.34}$	$46.01^{+0.59}_{-0.57}$
Enumeration	$100.00^{+0.00}_{-0.25}$	$85.38^{+1.68}_{-1.86}$	$88.78^{+1.56}_{-1.78}$	$55.60^{+3.36}_{-3.41}$	$51.59^{+2.16}_{-2.17}$	$76.27^{+0.92}_{-0.96}$
Enumeration + Geo.	$100.00^{+0.00}_{-0.25}$	$100.00^{+0.00}_{-0.25}$	$88.78^{+1.56}_{-1.78}$	$63.99^{+3.21}_{-3.34}$	$63.29^{+2.06}_{-2.11}$	$83.21^{+0.82}_{-0.87}$
TAILOR-G	$100.00^{+0.00}_{-0.25}$	$97.39^{+0.69}_{-0.92}$	$87.55^{+1.64}_{-1.85}$	$76.03^{+2.79}_{-3.04}$	$94.27^{+0.93}_{-1.09}$	$91.05^{+0.69}_{-0.77}$
TAILOR-F	$100.00^{+0.00}_{-0.25}$	$100.00^{+0.00}_{-0.25}$	$97.31^{+0.73}_{-1.00}$	$85.40^{+2.25}_{-2.58}$	$98.34^{+0.47}_{-0.65}$	$96.21^{+0.48}_{-0.57}$

Scenario-level intervals are 95% Wilson confidence intervals over 1,551, 1,532, 1,373, 822, and 2,043 requests for S1–S5, respectively. For Avg., lower and upper deviations are propagated separately through the unweighted mean of the five scenarios, treating scenarios as independent.

PSNR of 28.55 dB, below the lowest requested floor of 30 dB, so no request is accepted and its accepted-request fidelity metrics are undefined.

Enumeration. Enumeration evaluates all seven nonempty subsets of the three-fragment library at unit strength and fixed embedding order. Enumeration + Geo. extends the same subset sequence with compatible geometric stages. Each candidate is tested using the common recovery, image-quality, latency, and FPR criteria. With geometry, Enumeration draws on all 609 configuration-attack measurements.

C.4 Evaluation Metrics

We evaluate each method along three dimensions: whether it satisfies the complete deployment request, the fidelity of the resulting watermarked images, and the deployment latency required by the accepted configuration.

Request Satisfaction. A deployment request is satisfied only if the selected configuration passes all requested attacks during live evaluation while meeting the specified image-quality and latency requirements. We report satisfaction separately for each scenario and use the unweighted mean of the five scenario rates as the **Avg.** metric in Table 1.

Table 7 reports each satisfaction rate together with its 95% confidence interval. For TAILOR-F, the estimates are tightly concentrated in S1, S2, S3, and S5, while S4 has the largest uncertainty at $85.40^{+2.25}_{-2.58}\%$. The scenario-averaged satisfaction is $96.21^{+0.48}_{-0.57}\%$, where the uncertainty is propagated from the five scenario-level intervals. Overall, the high satisfaction of TAILOR-F is estimated with relatively low statistical uncertainty.

Paired Quality Comparison. Because different methods may satisfy different subsets of requests, raw fidelity averages are not directly comparable. For each baseline b, we therefore compare image fidelity on the requests jointly accepted by b and TAILOR. The PSNR-P column in Table 1 reports the mean PSNR of the baseline and TAILOR on this shared request set, using the same images for both methods. Each baseline induces a different intersection, so paired comparisons should be interpreted within each row rather than across rows. Figure 7 further shows request satisfaction and accepted-request PSNR across the five deployment scenarios, providing a scenario-level view of the coverage–fidelity trade-off.

Latency Accounting. We measure the deployment cost of each configuration as its embedding and verification latency under the requested attacks. All measurements use warm, resident models at batch size one on NVIDIA A100-SXM4-80GB GPUs with CUDA synchronization. For configu-

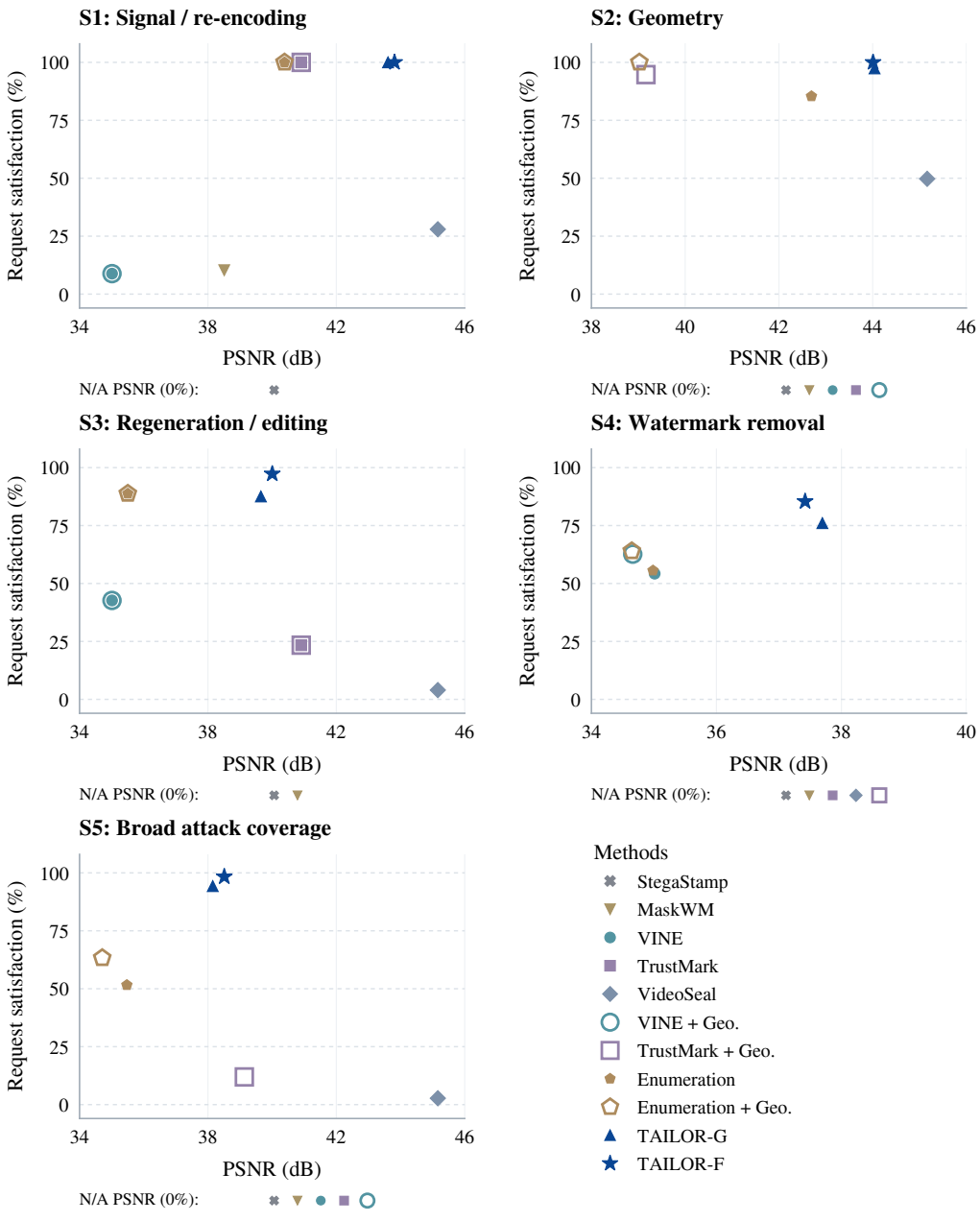


Figure 7: Request satisfaction and accepted-request PSNR across five deployment scenarios.

Table 8: Effect of the initial screening margin on request satisfaction and image fidelity.

		Request satisfaction (%) ↑						*Image fidelity*	
Method	m_0	S1	S2	S3	S4	S5	**Avg.**	PSNR ↑	SSIM ↑
TAILOR-G	0.04	91.23	94.19	86.96	71.17	70.19	82.75	**41.77**	**0.9931**
TAILOR-F	0.04	**100.00**	**100.00**	**97.31**	82.24	**98.34**	95.58	41.61	**0.9931**
TAILOR-G	0.06	**100.00**	97.39	87.55	76.03	94.27	91.05	40.91	0.9914
TAILOR-F	0.06	**100.00**	**100.00**	**97.31**	**85.40**	**98.34**	**96.21**	41.02	0.9923

Table 9: Image fidelity with continuous and grid strengths.

	Image fidelity	
Method	PSNR ↑	SSIM ↑
TAILOR-F	**40.66**	**0.9918**
Grid strengths	40.30	0.9914

ration c and deployment request $u = (\mathcal{A}_u, \alpha, q, t)$, the measured latency is

$$L_{\text{meas}}(c, u) = \max_{a \in \mathcal{A}_u} \frac{1}{N_a} \sum_{i=1}^{N_a} \left(t_i^{\text{embed}}(c) + t_{i,a,\alpha}^{\text{decode}}(c) \right), \tag{17}$$

where $\mathcal{A}_u$ is the set of attacks requested by u, N_a is the number of evaluated images under attack a, $t_i^{\text{embed}}(c)$ is the time required to embed configuration c into image i, and $t_{i,a,\alpha}^{\text{decode}}(c)$ is the verification time for the resulting image under attack a and FPR budget α. The outer maximum takes the most expensive requested attack condition as the deployment latency. Model loading, image I/O, and attack execution are excluded from the latency budget.

D Additional Ablation Studies

We examine the screening margin, fragment interference, recovery-interaction model, and strength discretization.

D.1 Impact of the Screening Margin

We evaluate how the initial rate-screening margin affects the trade-off between request coverage and image fidelity. Using the same 7,321 requests and fixed live-validation criteria, we compare $m_0 = 0.04$ and $m_0 = 0.06$. As shown in Table 8, increasing the margin raises TAILOR's scenario-averaged satisfaction from 95.58% to 96.21%, while mean PSNR over accepted requests decreases from 41.61 to 41.02 dB.

The coverage gain is concentrated in S4, where satisfaction increases from 82.24% to 85.40%, corresponding to 26 additional accepted requests among its 822 requests; the other four scenario rates remain unchanged. The greedy variant is more sensitive to the same change, increasing from 82.75% to 91.05% satisfaction while its mean PSNR decreases from 41.77 to 40.91 dB. These results show that a stronger screening margin can steer the search toward configurations that satisfy more difficult requests at some fidelity cost, while the full search remains comparatively stable across the two settings.

D.2 Continuous vs. Grid Strengths

We evaluate whether continuous strength optimization provides an advantage over selecting only the strengths explicitly measured during offline characterization. The grid variant restricts each fragment to the frozen response-curve knots while preserving all other configuration choices and the same live-validation procedure. Table 9 compares the two variants on the 1,061 requests accepted by both. Continuous optimization improves mean PSNR by 0.36 dB on this matched set, showing that interpolation between measured strengths enables TAILOR to find lower-distortion configurations than discrete strength selection alone.

D.3 Direct Fragment Superposition

We evaluate whether VINE, TrustMark, and VideoSeal remain independently decodable when embedded together, since complementary robustness is useful only if the fragments can coexist with limited interference. We compare standalone and sequentially composed watermarks on the same 100 512×512 images from the evaluation pool, containing 25, 25, 25, 15, and 10 images from the five source groups, respectively. We use the deployed VINE-R, TrustMark-B, and VideoSeal implementations with all embedding strengths fixed to one. The experiment covers all three standalone fragments, six ordered pairs, and six triple permutations, for a total of 1,500 configuration-image

Table 10: Independent raw-bit decoding after watermark superposition on 100 images.

	Bit accuracy (%)			*All selected fragments (%)*	
Embedding order	VINE	TrustMark	VideoSeal	$\geq 64/100$ bits	Exact words
V	99.97	-	-	100	97
T	-	100.00	-	100	100
S	-	-	99.59	100	83
$V \to T$	99.98	100.00	-	100	98
$V \to S$	99.96	-	99.48	100	76
$T \to V$	99.97	100.00	-	100	97
$T \to S$	-	99.96	99.58	100	78
$S \to V$	99.96	-	99.21	100	70
$S \to T$	-	99.98	96.95	100	28
$V \to T \to S$	99.95	99.97	99.52	100	75
$V \to S \to T$	99.95	99.98	96.78	100	27
$T \to V \to S$	99.96	99.96	99.51	100	70
$T \to S \to V$	99.97	99.96	99.27	100	62
$S \to V \to T$	99.96	99.99	95.54	100	21
$S \to T \to V$	99.97	99.98	95.72	100	21

measurements and 3,300 selected-fragment readouts. For each image, the same 100-bit codeword and fragment-specific key transformations are used across configurations. No attack, score fusion, or geometric recovery is applied, and every selected fragment is decoded independently from the composed image.

In Table 10, V, T, and S denote VINE, TrustMark, and VideoSeal, and arrows indicate embedding order. Bit accuracy is computed from the aligned LLR signs against the expected codeword, with zero scores counted as errors. Each mean is computed over all 100 images without filtering. We use the same 64-of-100 verification threshold for every configuration so that threshold differences cannot explain changes in recovery.

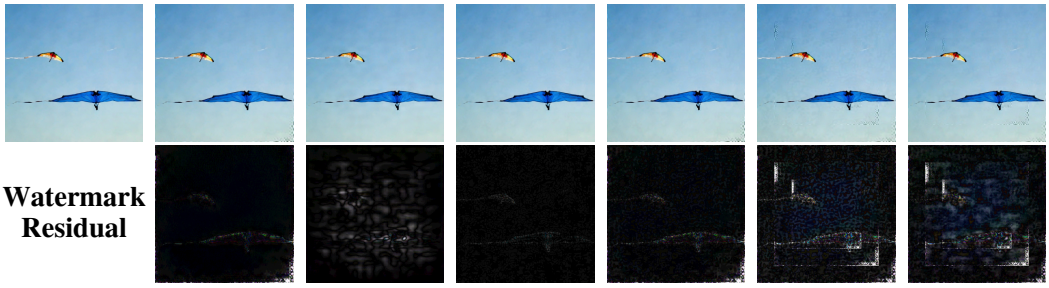
Standalone mean bit accuracy is 99.97% for VINE, 100.00% for TrustMark, and 99.59% for VideoSeal. Across all ordered pairs and triples, the corresponding composed readouts average 99.37%, compared with 99.85% for their standalone counterparts, a decrease of only 0.49 percentage points. The largest mean loss is 0.02 points for VINE and 0.04 points for TrustMark. VideoSeal is more order-sensitive: its loss remains below 0.08 points when embedded last in a triple but can reach 4.05 points under other orders. Nevertheless, every selected fragment remains above the verification threshold across all 100 images for every tested composition.

These results show that the selected fragments can coexist with low verification-level interference, while embedding order remains important, particularly for VideoSeal. Together with their complementary robustness across attack scenarios, this supports VINE, TrustMark, and VideoSeal as compatible building blocks for the fragment library of TAILOR.

D.4 RECOVERY-INTERACTION SENSITIVITY

We further test how sensitive TAILOR is to explicit modeling of recovery interactions between composed fragments. The NoRecoveryInteraction variant removes pairwise recovery corrections and composition-conditioned stage adjustments from the surrogate, while retaining live validation, local refinement, and margin fallback. At $m_0 = 0.04$, it satisfies 1,060 of the 1,111 requests, compared with 1,061 for the full method (95.41% versus 95.50%). On the 1,060 jointly accepted requests, mean PSNR is 40.690 dB for the variant and 40.664 dB for the full method, with a median paired difference of zero. The negligible change in both satisfaction and fidelity shows that the complete pipeline is largely insensitive to explicit recovery-interaction modeling, consistent with the low interference observed in direct superposition. Together with their complementary robustness, this supports VINE, TrustMark, and VideoSeal as sufficiently compatible building blocks for the fragment library of TAILOR.

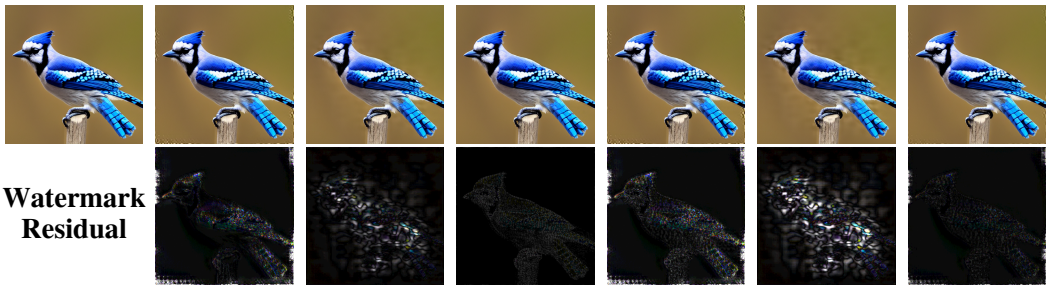
S2 · 10 Attacks | FPR ≤ 7.28×10^{-12} | PSNR ≥ 32 dB | Latency ≤ 1,000 ms



	Original	VINE	TrustMark	VideoSeal	Enumeration	TAILOR-G	TAILOR-F
Attacks Passed		5 / 10	7 / 10	7 / 10	7 / 10	10 / 10	10 / 10
PSNR ≥ 32 dB		35.01 dB	40.91 dB	45.16 dB	34.58 dB	33.41 dB	34.09 dB
Latency		63.0 ms	16.7 ms	13.5 ms	76.1 ms	247.6 ms	361.6 ms
FPR Bound		6.55×10^{-12}	6.55×10^{-12}	6.55×10^{-12}	3.91×10^{-12}	5.63×10^{-12}	5.67×10^{-12}
Request Outcome		**Rejected**	**Rejected**	**Rejected**	**Rejected**	**Accepted**	**Accepted**

Figure 8: Watermarked images, residuals, and constraint outcomes for a geometric-transformation request (S2).

S3 · 9 Attacks | FPR ≤ 10^{-1} | PSNR ≥ 34 dB | Latency ≤ 1,000 ms



	Original	VINE	TrustMark	VideoSeal	Enumeration	TAILOR-G	TAILOR-F
Attacks Passed		8 / 9	8 / 9	8 / 9	9 / 9	9 / 9	9 / 9
PSNR ≥ 34 dB		35.01 dB	40.91 dB	45.16 dB	34.58 dB	36.28 dB	42.66 dB
Latency		62.9 ms	16.7 ms	13.6 ms	76.1 ms	18.0 ms	76.1 ms
FPR Bound		9.67×10^{-2}	9.67×10^{-2}	9.67×10^{-2}	8.53×10^{-2}	9.67×10^{-2}	8.53×10^{-2}
Request Outcome		**Rejected**	**Rejected**	**Rejected**	**Accepted**	**Accepted**	**Accepted**

Figure 9: Watermarked images, residuals, and constraint outcomes for a regeneration-and-editing request (S3).

E Qualitative Request Examples

We provide qualitative examples to illustrate both how TAILOR compares with alternative configurations and how its selected composition changes with the deployment request.

Figures 8–10 compare watermarked images, amplified residuals, and constraint outcomes for representative requests from S2–S4. Each example preserves the original request budgets and tested configurations.

Figures 11, 12, and 13 trace three additional requests from specification to the configuration accepted by TAILOR. Each figure shows the requested attacks and budgets, the selected fragments and recovery stage, and the corresponding live-validation results. For multi-fragment configurations, the per-fragment columns report the pass rate of each fragment's own readout at the configuration threshold, illustrating how different fragments contribute coverage to different attacks. As the requests become broader, TAILOR moves from a single TrustMark fragment, to a VideoSeal–VINE composition, and finally to all three fragments with scale search. Residuals use the same 12× display gain, and latency and FPR are evaluated using the protocol above.

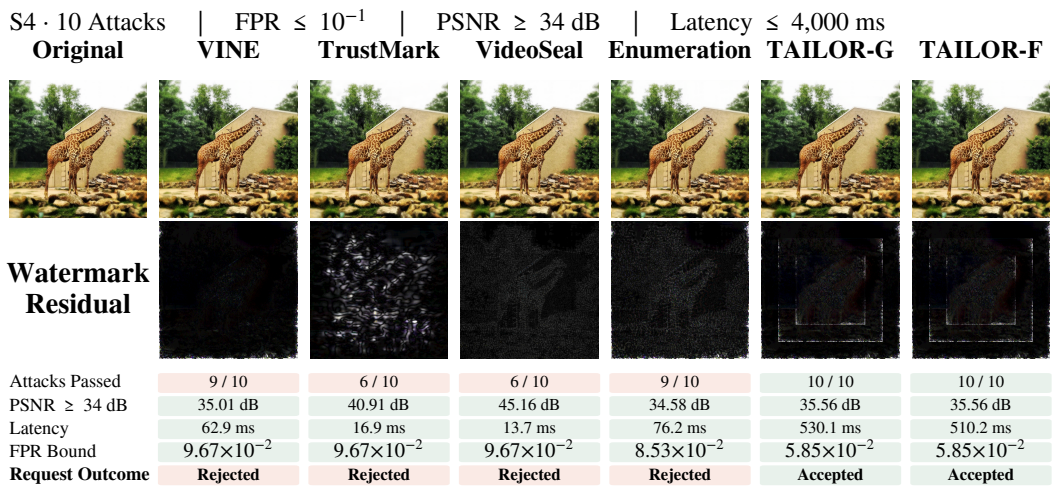


Figure 10: Watermarked images, residuals, and constraint outcomes for a watermark-removal request (S4).

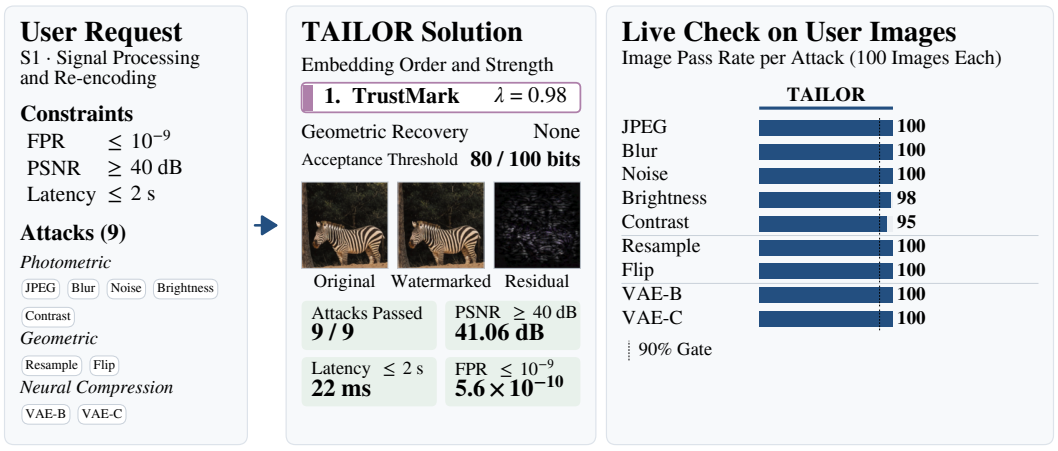


Figure 11: A signal-processing request (S1) answered by a single fragment.

F LIMITATIONS

TAILOR relies on offline characterization and can only optimize over watermarks, attacks, and recovery mechanisms represented in its performance database. Supporting new components or previously unseen attacks therefore requires additional profiling and database updates before they can be incorporated into configuration selection. In addition, the search is limited by the coverage and accuracy of the offline surrogate model; feasible configurations that are poorly represented by the database may be missed before live validation. Live calibration helps address prediction errors by providing measurement-backed corrections when offline predictions do not transfer accurately to the target deployment.

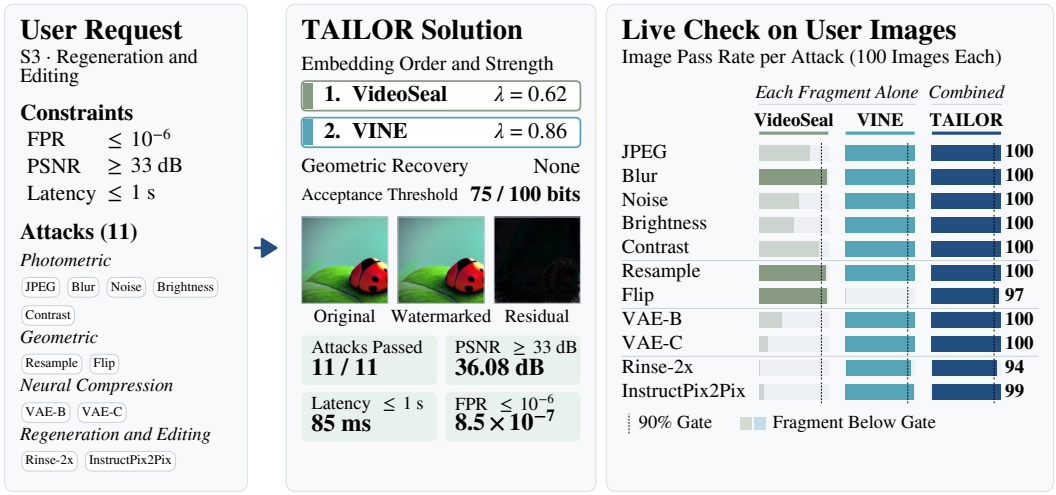


Figure 12: A regeneration-and-editing request (S3) answered by a complementary fragment pair.

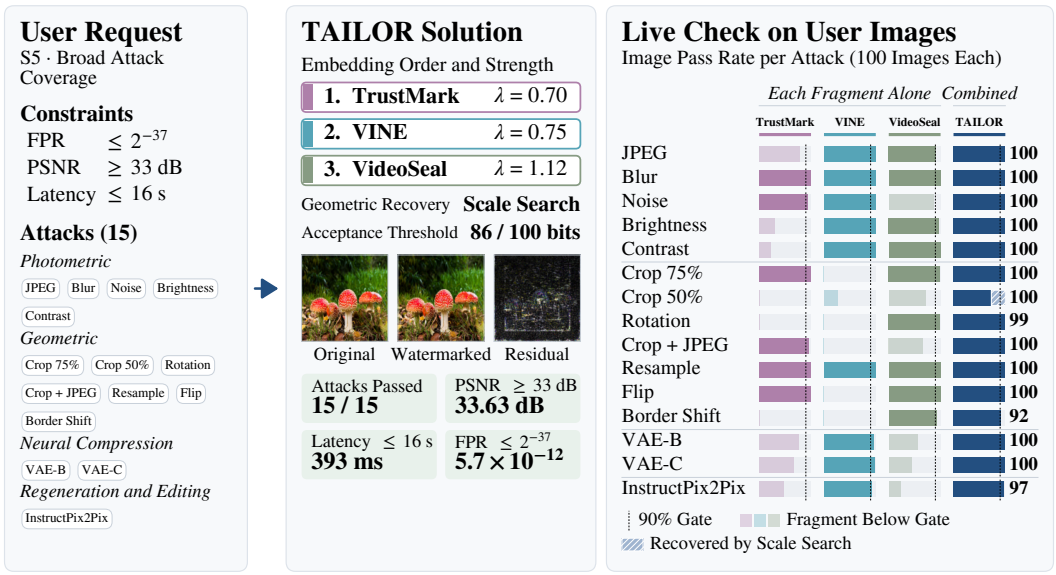


Figure 13: A broad-coverage request (S5) answered by three fragments with scale recovery.