Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Codebook-constrained soft decoding improves message recovery and attribution in localized watermarks

Hyeong Gyun Yun

xprise@g.hongik.ac.kr

Hongik University

Hye Young Kim

Hongik University

Article

Keywords:

Posted Date: November 16th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11435462/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Codebook-constrained soft decoding improves message recovery and attribution in localized watermarks

Hyeong Gyun Yun[1] and Hye Young Kim[2*]

[1]Department of Games, Graduate School, Hongik University, Sejong, Republic of Korea.

[2*]Department of Game Software, School of Games, Hongik University, Sejong, Republic of Korea.

*Corresponding author(s). E-mail(s): hykim@hongik.ac.kr;
Contributing authors: xprise@g.hongik.ac.kr;

Abstract

Localized image watermarking embeds several messages in different image regions, so composite content can be traced to several users. Watermark Anything (WAM) writes raw 32-bit messages and reports the bit accuracy of the clusters it finds; exact message recovery and the control of false attribution to registered users were not measured. Without retraining WAM, we restricted messages to a 16-bit error-correcting codebook, decoded each detected region by soft maximum-likelihood search over all 65,536 codewords, and accepted an identity only if it was registered and scored above a threshold calibrated on held-out images. In a pre-specified evaluation on 3,000 test images, exact recovery under a horizontal flip with contrast change rose from 0.508 to 0.944. A random codebook of the same size performed identically, indicating the gain comes from codeword separation and soft decoding rather than algebraic code structure. Calibrated attribution reached a correct-attribution rate of 0.802 with false attribution below 0.32% (upper confidence bound), versus 0.579 for a calibrated bit-tolerance rule. WAM's metric overestimated exact recovery (0.961 versus 0.508), and nearby messages were merged. An attacker knowing the public codebook can impersonate users; with secret codeword assignment no impersonation was observed, but forged regions were attributed to other users.

Keywords: Digital watermarking, Localized watermarking, Soft-decision decoding, Error-correcting codes, Conformal calibration, Provenance attribution

Introduction

Invisible watermarks are used to label content generated by artificial intelligence (AI) and to trace the provenance of images[1–3]. Most learned watermarking methods embed one message in the whole image and decode it globally[4–11], which makes decisions difficult when only part of an image carries a watermark or when content from several sources is combined. Watermark Anything (WAM)[12] treats watermarking as a segmentation problem: it predicts for every pixel whether the pixel is watermarked and which 32-bit message it carries, and it groups the pixel-wise bit strings with density-based spatial clustering of applications with noise (DBSCAN)[13, 14], using a Hamming radius $\varepsilon = 1$ and at least 1,000 pixels per cluster, to read several messages from one image. Each decoded message must then be linked to a registered user identifier (ID), and accepting a wrong ID (a false attribution) is a security failure.

The WAM paper and its public code leave four aspects of multi-message use unaddressed. First, message-level recovery: messages are raw 32-bit strings without error correction, and the centroid of each DBSCAN cluster is read directly as the message, so a few bit errors invalidate the message. Second, control of false attribution: every 32-bit string is a valid output, so no procedure prevents unregistered messages or damaged regions from being accepted as registered IDs. Third, nearby messages: DBSCAN links pixels whose bit strings differ by at most one bit, yet WAM was evaluated only with uniformly random messages, so the merging of similar messages placed in one image was not measured. Fourth, the evaluation metric: the multi-message bit accuracy is computed only over the clusters that DBSCAN finds (Sec. 5.5 of ref. [12]), so missed messages and messages with a few wrong bits remain hidden.

We kept the WAM model and its multi-message scene construction unchanged and modified only the encoding before embedding and the processing after detection (Fig. 1). Messages are restricted to codewords of an extended Bose–Chaudhuri–Hocquenghem (BCH) code[15, 16] with 16 information bits (BCH16), and each DBSCAN region is decoded by soft maximum-likelihood (ML) search over all 65,536 codewords using the mean bit logits of the region (Algorithm 1). A decoded ID is accepted only if it is registered and its score exceeds a threshold calibrated on held-out images with a conformal rank rule[17–19] (Algorithm 2). We fixed the hypotheses, decision rules and execution protocol before testing and recorded them by cryptographic hash, following the logic of pre-registration[20] without an external registry, and tested four hypotheses (H1–H4; Table 1) on 3,000 test images of the Common Objects in Context (COCO) dataset[21] with the checkpoint used for evaluation in the WAM paper.

Figure 1 follows one reserve image through embedding (a–e) and decoding with attribution (f–j). Five registered IDs are mapped to BCH16 codewords and embedded by the unchanged WAM embedder; after distortion, WAM's detection (f) and DBSCAN regions (g) are unchanged, and our modifications start from the mean bit logits of each region (h), which are compared with all 65,536 codewords (i) and accepted only for a registered ID above the calibrated threshold (j). All five IDs were accepted, whereas the raw WAM readout of the same image carrying random messages recovered 4 of the five messages exactly.

Our contribution is not a new decoding algorithm; soft-decision and maximum-likelihood decoding of block codes are standard in coding theory[22, 23]. Instead, we measure, in controlled experiments with pre-specified decision rules, what this standard technique and a calibrated acceptance rule achieve when attached to localized multi-message watermarking without retraining, and why. We (1) quantify the gain in exact message recovery and separate its cause—the separation of codewords over all 32 bits and soft decoding, not the algebraic structure of the code or a smaller candidate set; (2) provide a calibrated procedure for attribution to a registry of many users and report its effect under different threat models, calibration samples and registry sizes; (3) quantify DBSCAN merging as a function of message distance, including the non-trivial range of distances 3 and 4; and (4) quantify how much, and why, the WAM metric overestimates exact recovery. Nor do we claim novelty for conformal calibration, for the distinction between bit and message accuracy, or for a panoptic-quality-style metric.

Related work. EditGuard[24] and OmniGuard[25] combine tamper localization with copyright protection, with one copyright message per image. MaskWM[26] does not use DBSCAN; it splits a predicted watermark mask into connected components and extracts one message per component (up to five watermarks of 5% area each), without studying message distance or error correction. Among learned image watermarks, StegaStamp[5] uses BCH codes with hard decoding of a single global message, TrustMark[10] suggests BCH error correction but reports its results without it, and HiDDeN[4], Stable Signature[9] and Video Seal[11] report their main results without error correction. Soft-decision error correction of neural watermark outputs has been reported for text watermarks of large language models (LLMs): CORE-BREW[27] decodes token-level log-likelihood ratios and WeaveMark[28] uses soft-decision codes. We found no study that decodes localized multi-message image watermarks region by region over a full codebook from pixel logits. For attribution, Stable Signature and Three Bricks[29] set thresholds from theoretical false-positive rates under binomial or gamma models, WOUAF[30] addresses attribution to many users, model fingerprints have been used to attribute generated images to their source models[3, 31], and conformal calibration has been used for single-hypothesis detection[32, 33]; we found no model-free calibration for attribution to a registry of N users. Collusion-resistant fingerprinting codes[34, 35] address a different, adversarial threat model. Studies of multi-bit LLM watermarks report bit accuracy and message-level match rates separately[28] and criticize multi-bit evaluation metrics[36]; panoptic quality[37] combines detection and accuracy in segmentation, but applied to bit accuracy it does not reflect decoding success (see Results).

Results

Pre-specified evaluation

The primary H1–H4 evaluation was run once, after the protocol had been locked, on 3,000 test images (COCO val2017 images 1,500–4,499 in name order) with the WAM checkpoint wam_coco. Each image contained $k = 1$–5 messages in non-overlapping squares covering 10% of the image each, as in the multi-message evaluation of WAM.

The main outcome is the exact message recovery rate

$$\mathrm{ER} = \frac{1}{|\mathcal{M}|} \sum_{j \in \mathcal{M}} \mathbf{1}[\exists\, r \in \mathcal{R}_j :\ \hat{u}_r = u_j]\,, \tag{1}$$

where $\mathcal{M}$ is the set of embedded messages whose visible area is at least 1,000 pixels, u_j is the embedded message (the 32-bit string for raw decoding or the ID for codebook decoding), $\hat{u}_r$ is the output of decoded region r, and $\mathcal{R}_j$ is the set of regions assigned to message j; as in WAM, each region is assigned to the message $a(r)$ whose visible area it overlaps most. A message without any assigned region counts as a failure, so equation (1) penalizes both missed messages and messages with wrong bits.

Table 1 gives, for each hypothesis, the statement, the decision rule fixed before any test image was measured, the result and the verdict. Each rule compares two decoders or message sets on the same images by a paired difference with a 98.75% image-paired bootstrap interval[38] (equation (15); Bonferroni correction over four hypotheses), and a hypothesis is supported when the lower end of the interval is above zero; H2 additionally requires one-sided Clopper–Pearson upper bounds[39] (equation (16)) on false attribution of at most 1% for both attribution methods, and all conditions of a hypothesis must hold, as in an intersection–union test[40]. All four hypotheses were supported. H3 part (1) and H4 follow almost from the definitions of DBSCAN and of the WAM metric, so the following sections report the size of each effect and its causes rather than the verdict alone.

Codebook-constrained soft decoding recovers more messages exactly

Under horizontal flip plus contrast 1.5, the distortion used for the decision, exact recovery (equation (1)) rose from 0.508 with WAM's raw decoding to 0.944 with BCH16 soft decoding (+0.435 [+0.424, +0.447]). Figure 2a extends this comparison to all 26 WAM evaluation distortions. Each row shows the raw recovery (grey) and the BCH16 recovery (blue) of one distortion, joined by a line whose length is the gain, and the rows are sorted by raw recovery. The gain is small at the top, where raw decoding already succeeds, and largest in the middle and lower rows, where the watermark is still detected but several bits are corrupted: rotation by −10 degrees (+0.751), rotation by +10 degrees (+0.751), JPEG 80 (+0.721), perspective 0.1 (+0.717). At the bottom, both decoders fail under the two compound distortions (JPEG 80 + brightness 1.5 + crop 0.5, and h-flip + contrast 1.5 + JPEG 80) because the watermark signal is lost. The difference was significant in 25 of the 26 conditions after Holm correction[41] (filled circles), and 24 conditions showed a difference of at least 0.17. The public checkpoint wam_mit showed the same direction (+0.259 [+0.248, +0.271]). Reducing the payload increased recovery: 0.508 with 32 information bits (WAM), 0.909 with 21 bits (extended BCH(32,21), BCH21) and 0.944 with 16 bits (BCH16).

Figure 2b separates the source of this gain in the decision condition. Each row is a paired difference with its 98.75% interval, and each comparison removes one candidate explanation. A random codebook of 65,536 words (RND16) gave the same recovery as BCH16 (0.000 [−0.004, +0.003], red point on the zero line), so under this condition the

algebraic structure of BCH added nothing measurable (which does not make it irrelevant elsewhere). A 16-bit ID followed by a fixed 16-bit suffix (PAD16), which adds no distance between IDs and no error correction, improved on WAM by only +0.091 [+0.080, +0.102], and BCH16 exceeded it by +0.344 [+0.334, +0.355], so a smaller candidate set explains little of the gain. A repetition code (REP16), in which every codeword has 16 neighbours at distance 2, was lower than BCH16 (+0.104 [+0.097, +0.111]), and soft decoding added +0.053 [+0.048, +0.057] over hard decoding of the same codebook, with larger gains where bit errors were frequent (JPEG 80, +0.248; rotation by 10 degrees, +0.179). The PAD16 comparison was first measured on 500 reserve images (+0.342 [+0.317, +0.368]) and then reproduced on the test images under a separately locked plan. The locked H1 statement attributed the improvement to "candidate restriction and soft decoding" with "no difference from a random codebook of the same size". The first part was an interpretation from development data that Fig. 2b revises to codeword separation and soft decoding; the second part was confirmed (equivalence margin ±0.02, tested by interval inclusion[42]). The H1 verdict rests only on the primary comparison and is unaffected.

Calibrated registry attribution controls false attribution

The attribution test used mixed scenes with $k = 0$–5 messages and a registry U of 1,000 IDs; half of the embedded messages were registered, and half of the unregistered ones were random while the other half lay two bits away from a registered raw message (near-miss messages). For a set of n test scenes, the false-attribution rate and the correct-attribution rate are

$$\begin{aligned} \mathrm{FA} &= \frac{1}{n}\sum_{t=1}^{n} \mathbf{1}[\exists\, r :\ \hat{u}_r \in U \setminus \mathcal{E}_t,\ s_r > \tau]\,, \\ R &= \frac{1}{|\mathcal{M}_U|}\sum_{j \in \mathcal{M}_U} \mathbf{1}[\exists\, r \in \mathcal{R}_j :\ \hat{u}_r = u_j,\ s_r > \tau]\,, \end{aligned} \tag{2}$$

where $\mathcal{E}_t$ is the set of IDs embedded in scene t, $\hat{u}_r$ and s_r are the decoded ID and score of region r, τ is the acceptance threshold and $\mathcal{M}_U$ is the set of embedded, registered and eligible messages. A scene therefore counts as a false attribution as soon as one region is accepted with a registered ID that was not embedded, while R counts only acceptances with the correct ID.

Figure 3a shows the trade-off between the two quantities of equation (2). Each curve traces R against FA on the test scenes as the acceptance threshold of one method is varied (descriptively, without changing any decision), and the diamonds mark the thresholds that were calibrated beforehand on 1,000 calibration images. At every false-attribution level up to 2%, method C (full-codebook soft decoding, registry check and calibrated threshold) attains a higher R than method A (WAM raw decoding with a calibrated Hamming tolerance t) and method B (soft decoding within the registry only). At the calibrated operating points, C made 3/3,000 false attributions (FA = 0.10%; one-sided 98.75% Clopper–Pearson upper bound 0.32%) with $R = 0.802$, whereas A made 15/3,000 (FA = 0.50%; upper bound 0.87%) with $R = 0.579$, both

pooled over the 3,442 registered eligible messages (equation (2)), a pooled difference of +0.223 [+0.203, +0.243] (image-level bootstrap, post hoc). The locked H2 statistic, the mean per-image difference Δ_{img} over the 1,915 images with at least one registered eligible message (Methods), was +0.213 [+0.193, +0.233] (per-image means 0.792 and 0.579). All false attributions of A came from near-miss messages, which forced its tolerance to $t = 1$. B also controlled false attribution (4/3,000, FA = 0.13%, upper bound 0.38%) but required a strict threshold that limited R to 0.152 (pooled difference $R(C) - R(B) = +0.650[+0.625, +0.675]$; per-image mean difference +0.637 [+0.611, +0.662]). B and C differ in message representation, candidate set and score, so this compares complete pipelines; one plausible contributor is that C rejects unregistered messages and damaged regions that decode to unregistered codewords. No method produced false attributions in unwatermarked images (483 scenes) or under signal-loss distortions (299 scenes).

Figure 3b tests the sensitivity of this advantage to near-miss messages (E2) by excluding, in the existing calibration and test records, the null scores from regions of near-miss messages; no new scenes were generated, so these messages remained embedded and could still affect detection and clustering. With both methods recalibrated by the same rule, the tolerance of A widened to $t = 3$ and $R(A)$ rose to 0.717, so the advantage of C shrank but remained (pooled difference +0.085 [+0.071, +0.099]; per-image mean difference +0.083 [+0.068, +0.098]).

The threshold of C rests on only 4 calibration scenes that produced a null score, because it equals the second largest of them. When we resampled the calibration scenes 2,000 times, 9.5% of the resamples had fewer than two null scores and therefore no finite threshold (every registered decoding accepted); in the others the threshold varied between 37.2 and 71.2 (5th–95th percentiles). Including all resamples, the test false-attribution rate had a median of 0.10% and a 95th percentile of 0.30%, 0.0% of the resamples exceeded 1%, and R ranged from 0.757 to 0.813 (A: 0.443–0.579, with t switching between 0 and 1). To approximate the effect of the registry size N, we extrapolated the share of calibration and test scenes that contained a wrong decoding to a registered ID, $p_{1000} = 0.325\%$, as

$$p_N = 1 - (1 - p_{1000})^{N/1000}, \tag{3}$$

which assumes that wrong decodings fall uniformly on the codebook independently of N. Equation (3) gives 0.32%, 3.20% and 19.21% for N = 1,000, 10,000 and 65,536. Re-applying the same conformal rule to simulated calibration sets with these rates keeps false attribution controlled but raises the threshold, lowering the median R of C to 0.810, 0.763 and 0.727 (a simulation from 346 development null scores, not a measurement).

Nearby messages are merged by DBSCAN

For two messages embedded in one image, with visible pixel sets V_1 and V_2, we record a merge when a single DBSCAN region covers at least half of each:

$$\text{merge} = \mathbf{1}\left[\exists\, r:\ |R_r \cap V_1| \geq \tfrac{1}{2}|V_1| \text{ and } |R_r \cap V_2| \geq \tfrac{1}{2}|V_2|\right]. \tag{4}$$

Figure 4a plots the merge rate of equation (4) against the Hamming distance between the two raw messages, without distortion (green) and under h-flip + contrast 1.5 (purple); the dashed lines mark nearest BCH16 codeword pairs, which are 8 bits apart, and the crosses mark random pairs. Two messages at distance 1 fall into one cluster when no bit error occurs, as expected for $\varepsilon = 1$ (merge rate 0.962 without distortion and 0.804 with distortion; H3 part 1, +0.883 [+0.872, +0.893] against random pairs). The curves decrease with distance but stay above the dashed lines at distances 2–4. Figure 4b expresses this as paired differences from the BCH16 pairs: averaged over distances 3 and 4 and both distortions, raw pairs merged more often by +0.195 [+0.182, +0.206]. The difference almost vanished at distance 6 (+0.010 [+0.003, +0.016] without distortion, +0.034 [+0.023, +0.046] with distortion) and disappeared at distance 8, where raw pairs merged as rarely as the BCH16 pairs (0.002 and 0.015 versus 0.005 and 0.015); the reduction therefore comes from the distance rather than from the code itself. Under distortion, nearest BCH16 pairs merged slightly more often than random pairs.

The WAM metric overestimates exact recovery

WAM evaluates multi-message decoding by the bit accuracy of the clusters that DBSCAN finds,

$$\mathrm{BA} = \frac{1}{|\mathcal{F}|} \sum_{r \in \mathcal{F}} \frac{1}{32} \sum_{b=1}^{32} \mathbf{1}\left[\bar{m}_{r,b} = m_{a(r),b}\right], \qquad \Delta = \mathrm{BA} - \mathrm{ER}, \tag{5}$$

where $\mathcal{F}$ is the set of regions assigned to an embedded message, $\bar{m}_r$ is the centroid (majority bits) of region r and $m_{a(r)}$ is the 32-bit message assigned to it. BA is correct for what it measures but does not measure exact recovery; two properties make Δ positive almost by definition: missed messages do not enter BA, and a message with one wrong bit still contributes 31/32 to BA but 0 to ER. Under h-flip + contrast 1.5, BA was 0.961 while ER was 0.508 (Δ = +0.452 [+0.440, +0.464]).

Figure 5a shows that this gap is general: for every one of the 26 distortions, the point (BA, ER) lies below the dashed equality line, and the 17 conditions with BA of at least 0.90 span exact recovery from 0.173 to 0.808, so a high BA says little about whether messages are recovered. Figure 5b decomposes the outcome of every embedded message into exact recovery, found with 1–3 bit errors, found with 4 or more bit errors, and missed. In the decision condition, 6.1% of the failed messages were missed, 73.7% were found with 1–3 bit errors and 20.2% with at least 4 bit errors, so the hidden failures are mostly a few wrong bits, which the codebook corrects. Without distortion, 93% of the failures were 1–3-bit errors, whereas under JPEG 50, missed messages and errors of 4 or more bits dominated. A panoptic-quality-style metric built from detection quality and centroid bit accuracy was identical for the two decoders (0.945 and 0.945 under h-flip + contrast 1.5), because it does not reflect decoding success.

Embedding quality and computational cost

On 500 reserve images, the full-image peak signal-to-noise ratio (PSNR) of BCH16 embedding differed from that of raw embedding by −0.0003 [−0.0015, +0.0007] dB, inside the ±0.1 dB equivalence margin; the mean PSNR was 37.45 dB and the structural similarity index (SSIM)[43] 0.9676 for raw messages. The WAM paper reports 38.3 dB under its own protocol; our values were measured at 256 × 256 pixels and are not directly comparable. On an RTX 4090 graphics processing unit (GPU) with batch size 1 (median over 90 reserve images, h-flip + contrast 1.5), WAM embedding took 4.21 ms per message, detection 5.81 ms, DBSCAN 1.59 ms, and BCH16 soft decoding of all regions of an image 0.51 ms.

Forgery by an attacker who knows the codebook

We tested an attacker who has the same WAM embedder, the public codebook and decoding rule and the public ID of a target user, but neither a secret key nor access to the detector (500 reserve images, five targets per image, the calibrated threshold and registry of method C). Two ways of assigning IDs to codewords were compared: public assignment, in which the ID equals the codeword index (our default), and keyed assignment, in which a secret permutation maps IDs to codewords. Without the key, the attacker can only embed the public codeword of the target. Under keyed assignment such a forged region decodes to the ID $\pi_K^{-1}(\cdot)$ of an essentially random codeword, so the expected rate at which it is attributed to some registered user is

$$\mathbb{E}[F_{\text{unrelated}}] \approx \frac{N}{65{,}536}\, R_{\text{legit}}, \tag{6}$$

where R_{legit} is the rate at which a region carrying a codeword is decoded and accepted.

Table 2 reports the experiment for both assignments and two distortions. Its rows give legitimate attribution under each assignment and their difference, the success of targeted impersonation, the attribution of forged regions to unrelated registered users, and the expected value of equation (6). With public assignment, the attacker impersonated targets as often as legitimate users were attributed (2,363/2,500 under h-flip + contrast 1.5): calibrated acceptance controls only non-adversarial false attribution. With keyed assignment, no targeted impersonation was observed in the tested setting (0/2,500 regions; 0 of 500 images, one-sided 98.75% upper bound 0.87%; the region-level bound 0.18% assumes independent regions) and legitimate attribution was unchanged (+0.002 [−0.006, +0.011], within ±0.02). Forged regions, however, were attributed to unrelated registered users in 1.60% of cases with the primary key (median over 1,000 random keys, 1.44%), matching the 1.44% predicted by equation (6) and exceeding 1%. This is a per-region rate under attack, not the scene-level false-attribution rate that H2 bounds. Keyed assignment therefore turns impersonation into the framing of random users. Copying or replaying genuine watermarks[44], watermark removal[45, 46] and adaptive attacks[47] were not studied.

Discussion

Most failures of WAM's raw readout are errors of one to three bits (73.7% under h-flip + contrast 1.5). Restricting messages to a codebook lets such errors be corrected within the distance between codewords. Because a 16-bit ID with a fixed suffix barely improves recovery and a random codebook performs like BCH, what matters under our conditions is that codewords are far apart over all 32 bits; soft decoding with region-averaged logits adds a further gain. BCH remains a practical choice because its codebook can be generated rather than stored and its minimum distance is guaranteed.

For attribution, raw 32-bit decoding outputs every string, so accepting the nearest registered ID is vulnerable to unregistered messages close to registered ones. Decoding over the full codebook first and then checking registration can absorb unregistered messages and damaged regions into unregistered codewords (not isolated from the other differences to B). All methods used the same conformal rule at the nominal level $\alpha = 0.002$, and H2 required the upper confidence bounds on scene-level false attribution of A and C to be at most 1%; their observed rates were not equal (C 0.10%, A 0.50%); the per-image advantage was large when near-miss messages were present (+0.213) and persisted when their scores were excluded (+0.083). The calibrated threshold varies with the calibration sample because null scores are rare. False attribution remained controlled in our resampling, but deployments should use more calibration scenes or include enough signal-loss scenes. As the registry grows relative to the codebook, wrong decodings hit registered IDs more often and the threshold rises, so the codebook should be much larger than the expected number of users. Messages that may appear in the same image should be assigned at Hamming distance 6 or more, because merging is still substantial at distances 3 and 4. Evaluations of multi-message watermarks should report exact message recovery together with the causes of failure (missed messages, 1–3-bit errors and larger errors) rather than the bit accuracy of the clusters found.

Our decisions concern non-adversarial false attribution, that is, unregistered messages, damaged or signal-lost regions and near-miss messages. The multi-message evaluation of WAM considers only non-adversarial distortions, and we kept its conditions. A public codebook cannot resist forgery, which requires secret keys such as keyed codeword assignment or signatures. Our forgery experiment confirms this: public assignment allowed impersonation, and with keyed assignment no impersonation was observed, but forged regions were attributed to unrelated users at a per-region rate above 1%, because by equation (6) this rate grows with the ratio of registered IDs to codewords; keeping it below 1% would require at most about 655 registered IDs per 65,536 codewords without distortion, a larger codebook, or signatures (not tested). Robustness to watermark removal is determined by the WAM model, which we did not change, and the calibrated bound holds only for inputs distributed like the calibration data, not for inputs crafted by an attacker. We did not compare other localized watermarking models, because model differences would confound the effect of post-processing; EditGuard and OmniGuard carry one copyright message per image, and the proposed post-processing needs only region-wise bit logits, so it could in principle be applied to multi-message models such as MaskWM, which we did not test. A cryptographic signature is far longer than the 32-bit payload of a region, so it cannot be

embedded in the watermark itself; it would have to be kept outside the image, for example in a signed provenance record that the decoded ID points to. Future work should combine keyed assignment with a smaller registry ratio or such signed records and evaluate copying, replay and adaptive attacks[44, 47], examine message-level recovery under removal and regeneration attacks[45, 46], and reproduce the analysis on other localized multi-message models.

Further limitations apply. Decisions used one checkpoint (wam_coco; wam_mit agreed) and the 5,000 COCO val2017 images, whereas WAM used the first 10,000 images; scenes were WAM's 10% checkerboard at 256 × 256 pixels only, without irregular masks, other areas or high resolution. The payload drops from 32 to 16 bits (65,536 IDs). This is the price of redundancy: under h-flip + contrast 1.5 the nominal 2^{32} raw messages were recovered exactly in only 0.508 of cases, and BCH21 offers about two million IDs at 0.909 when more users are needed. Under strong compression combined with other distortions, none of the tested decoders recovered messages, and an oracle given the true regions (wam_mit, 200 development images) also showed bit errors at chance level. H2 targets false attribution in mixed scenes, a 1% guarantee per condition group was not tested, and the result depends on the near-miss definition and on a registry of 1,000 IDs; the effect of the registry size is an approximation. Nearest BCH16 pairs were not equivalent to random pairs within ±0.02 under distortion. The equal-capacity baseline, image quality, timing and forgery results are supplementary analyses rather than locked decisions, and timing was measured on one GPU with batch size 1.

In summary, keeping the multi-message conditions of WAM and changing only the encoding and post-processing raised exact recovery from 0.508 to 0.944 without retraining (h-flip + contrast 1.5) and raised registry attribution from 0.579 to 0.802 with an upper confidence bound on false attribution of 0.32%. The gain stems from well-separated codewords and soft decoding, with unchanged image quality and small computational cost. Messages placed in one image should be far apart, multi-message watermarks should be evaluated at the message level, and calibrated attribution must be complemented by keyed assignment and further safeguards against forgery.

Methods

Watermarking model and scenes

We used WAM[12] at commit 2c08af04 without modification, with the checkpoint used for evaluation in the WAM paper (wam_coco) for all decisions and the public checkpoint wam_mit as a secondary check, and kept the embedding strength of the released configuration (scaling_w 2.0, scaling_i 1.0, just-noticeable-difference attenuation[48]). Scenes followed the multi-message evaluation of WAM: images were resized to 256 × 256 pixels, the same image was passed to the embedder once per message, and each watermarked square was pasted onto the original,

$$y = \Big(1 - \sum_{j=1}^{k} M_j\Big) \odot x + \sum_{j=1}^{k} M_j \odot E\big(x, c(u_j)\big), \tag{7}$$

where x is the image, E is the WAM embedder, $c(u_j)$ is the codeword (or raw message) of the j-th ID, M_j is the binary mask of the j-th square and $\odot$ is pixel-wise multiplication. Equation (7) is the composite of Fig. 1c; only the message fed to E differs between raw and codebook embedding. Each region is a square of 10% of the image area; the squares occupy non-overlapping cells of a 3×3 checkerboard, and the number of messages is $k = 1$–5. Distortions were all evaluation settings of Appendix D.4 of the WAM paper, the compound distortion JPEG 80 + brightness 1.5 + crop 0.5 and the two compound distortions of Sec. 5.5 (h-flip + contrast 1.5, and the same followed by JPEG 80), 26 conditions in total, applied with the augmentation modules of the WAM code; geometric transforms were applied to the ground-truth masks as well. Messages whose visible area was smaller than 1,000 pixels (the DBSCAN minimum) were excluded from the decisions and counted separately.

Codebook and decoding

Sixteen-bit IDs were encoded with the extended BCH(32,16) code: the BCH(31,16) code with error-correcting capability 3 over GF(2^5) with primitive polynomial x^5+x^2+1, extended by an overall parity bit, which gives minimum distance 8. We generated all 65,536 codewords and verified the minimum distance. The extended BCH(32,21) code (minimum distance 6, about two million IDs) served for comparison. Detection, region formation and decoding follow Algorithm 1. The WAM extractor returns a watermark logit p_i and 32 bit logits $L_i \in \mathbb{R}^{32}$ for every pixel i; pixels with $\sigma(p_i) > 0.5$ are clustered by WAM's DBSCAN on their bits $\mathbf{1}[L_i > 0]$ (Hamming $\varepsilon = 1$, minimum 1,000 pixels). For each region R_r we average the bit logits,

$$l_r = \frac{1}{|R_r|} \sum_{i \in R_r} L_i, \tag{8}$$

and select the codeword with the largest correlation,

$$\hat{c}_r = \arg\max_{c \in \mathcal{C}} \langle c, l_r \rangle, \qquad s_r = \langle \hat{c}_r, l_r \rangle, \qquad \hat{u}_r = \mathrm{ID}(\hat{c}_r), \tag{9}$$

where codewords are written as $c \in \{-1, +1\}^{32}$ and s_r is the score used for attribution. Equation (8) keeps the confidence of every pixel instead of the majority vote, and equation (9) is evaluated for all 65,536 codewords as one $65{,}536 \times 32$ matrix product on the GPU. The rule is maximum-likelihood under a simple model of the averaged logits, $l_r = a\, c^* + e$ with gain $a > 0$ common to all candidate codewords, transmitted codeword c^* and noise $e \sim \mathcal{N}(0, \sigma^2 I_{32})$ with zero mean and a common variance $\sigma^2 > 0$ (an assumption of this simplified model, not verified on the data): because every codeword has the same norm $\|c\|^2 = 32$,

$$\|l_r - a\, c\|^2 = \|l_r\|^2 - 2a\langle c, l_r \rangle + 32a^2, \tag{10}$$

so the codeword that minimizes the squared distance in equation (10), and hence maximizes the Gaussian likelihood, is the one that maximizes the correlation in equation (9)

Algorithm 1 Codebook-constrained soft decoding for WAM multi-message watermarks

Require: image x; IDs $u_1, \ldots, u_k$ (16-bit, $k \le 5$); WAM embedder E and extractor D (unchanged); region masks $M_1, \ldots, M_k$ (WAM §5.5, 10% squares); codebook $\mathcal{C}$ = extended BCH(32,16), $|\mathcal{C}| = 65{,}536$, $d_{\min} = 8$, codewords $c \in \{-1, +1\}^{32}$

Ensure: regions R_r with decoded ID $\hat{u}_r$ and score s_r

Embedding

for $j = 1, \ldots, k$ **do**
  $w_j \leftarrow E\big(x, c(u_j)\big)$
end for
$y \leftarrow x \odot (1 - \sum_j M_j) + \sum_j w_j \odot M_j$

Decoding (y': y after any distortion)

$(p, L) \leftarrow D(y')$ ▷ p: watermark logits, $L \in \mathbb{R}^{32 \times H \times W}$: bit logits
$P \leftarrow \{i : \sigma(p_i) > 0.5\}$; $b_i \leftarrow \mathbf{1}[L_i > 0]$ for $i \in P$
$\{R_r\} \leftarrow \mathrm{DBSCAN}\big(\{b_i\};\ \text{Hamming } \varepsilon = 1,\ \text{min_samples} = 1000\big)$ ▷ WAM, unchanged
for each region R_r **do**
  $l_r \leftarrow \frac{1}{|R_r|} \sum_{i \in R_r} L_i$
  $\hat{c}_r \leftarrow \arg\max_{c \in \mathcal{C}} \langle c, l_r \rangle$; $s_r \leftarrow \langle \hat{c}_r, l_r \rangle$ ▷ one $65{,}536 \times 32$ product
  $\hat{u}_r \leftarrow \mathrm{ID}(\hat{c}_r)$
end for
return $\{(R_r, \hat{u}_r, s_r)\}$

regardless of a. For comparison, WAM raw decoding reads the centroid $\bar{m}_r$ (majority bits of the region) directly as the 32-bit message, and hard decoding selects the codeword nearest to the centroid,

$$\hat{c}_r^{\,\mathrm{hard}} = \arg\min_{c \in \mathcal{C}} d_H(c, \bar{m}_r)\,, \tag{11}$$

where d_H is the Hamming distance; equation (11) discards the per-bit confidence that equation (9) uses.

Algorithm 1 lists the complete procedure. Its embedding part implements equation (7) and its decoding part equations (8) and (9); only the encoding before the embedder and the steps after DBSCAN differ from WAM, so the method needs no retraining.

Registry attribution and calibration

Method C (Algorithm 2) accepts the ID of a decoded codeword only if the ID is registered and the maximum correlation score exceeds a threshold τ. For each calibration scene t with embedded IDs $\mathcal{E}_t$, the null score is the highest score of a region decoded to a registered ID that was not embedded,

$$z_t = \max\{s_r : \hat{u}_r \in U \setminus \mathcal{E}_t\} \quad (z_t = -\infty \text{ if the set is empty}), \tag{12}$$

so that a scene produces a false attribution exactly when $z_t > \tau$ (equation (2)). With $n = 1,000$ calibration scenes and $\alpha = 0.002$, a score s is accepted if the conformal p-value $(1 + \#\{t : z_t \geq s\})/(n+1)$ is at most α, which yields the threshold

$$\tau = z_{(m+1)}, \qquad m = \lfloor \alpha(n+1) \rfloor - 1, \tag{13}$$

where $z_{(m+1)}$ is the $(m+1)$-th largest finite null score ($\tau = -\infty$ if there are fewer). Here $m = 1$, so τ is the second largest null score, $\tau_C = 47.0758$. If the null score z_{new} of a new scene is exchangeable with the calibration scores, its rank among the $n+1$ scores is uniform when ties are broken at random; because many null scores are $-\infty$ and acceptance requires $s_r > \tau$ strictly, ties can only lower the probability below, so

$$\Pr\left(z_{\text{new}} > \tau\right) \leq \frac{m+1}{n+1} \leq \alpha. \tag{14}$$

Equation (14) is the false-attribution guarantee of method C. It is marginal, that is, averaged over calibration sets and new scenes rather than conditional on the realized threshold; it requires no model of the score distribution, but it holds only for scenes distributed like the calibration scenes and not for inputs crafted by an attacker. The calibration level $\alpha = 0.002$ is stricter than the 1% criterion of H2, which we checked separately on the test scenes with the one-sided Clopper–Pearson upper bound of equation (16); the reported bounds such as 0.32% are these test-set confidence bounds, not quantities given by the conformal rule. Method A, the WAM-style rule, decodes the raw message and accepts the nearest registered ID within Hamming tolerance t, with t calibrated by the same rule (score equal to minus the distance), which gave $t = 1$. Method B soft-decodes the raw message within the registry only (score equal to the best minus the second-best correlation), with $\tau_B = 90.50$ from the same rule. The registry consisted of 1,000 BCH16 IDs and 1,000 random 32-bit messages (seed 20261100), identical for calibration and testing.

Algorithm 2 lists the two stages of method C. Calibration computes the null scores of equation (12) on the calibration scenes and the threshold of equation (13); attribution accepts the decoded ID of a region only if $\hat{u}_r \in U$ and $s_r > \tau$, which is the acceptance condition counted in equation (2).

Comparison codebooks

To separate the source of the H1 gain, we used a random codebook of 65,536 random 32-bit words (RND16, minimum distance 1), a repetition code that repeats the 16 information bits twice (REP16, minimum distance 2), a 32-bit word formed by the 16-bit ID and a fixed 16-bit suffix common to all messages (PAD16, minimum distance 1; the suffix adds no distance between IDs and therefore no error correction for the ID bits) and the BCH21 codebook. All were decoded with the same soft decoder; PAD16 was also decoded by hard decision. No two BCH16 codewords are closer than 8 bits and very few random codewords are, whereas each REP16 codeword has 16 neighbours at distance 2 and PAD16 codewords differ only in their 16 ID bits.

Algorithm 2 Calibrated registry attribution

Require: registry U (N = 1,000); calibration scenes $t = 1, \ldots, n$ with known embedded IDs (n = 1,000); level $\alpha = 0.002$

Ensure: threshold τ; accepted IDs of a new image

Calibration

for $t = 1, \ldots, n$ **do**
  run Algorithm 1 $\rightarrow \{(\hat{u}_r, s_r)\}$
  $z_t \leftarrow \max\{s_r : \hat{u}_r \in U,\ \hat{u}_r$ not embedded in scene $t\}$ ▷ $-\infty$ if empty
end for
$m \leftarrow \lfloor \alpha(n+1) \rfloor - 1$; $\tau \leftarrow$ the $(m+1)$-th largest finite z_t ▷ $-\infty$ if fewer

Attribution

run Algorithm 1 on the new image $\rightarrow \{(R_r, \hat{u}_r, s_r)\}$
accept $\hat{u}_r$ **iff** $\hat{u}_r \in U$ **and** $s_r > \tau$

Marginal guarantee (exchangeable, non-adversarial scenes; ties broken conservatively): Pr(false attribution) $\leq (m+1)/(n+1) \leq \alpha$

Data and scene designs

The 5,000 COCO val2017 images[21] were split by name order into development (images 0–499), calibration (500–1,499), test (1,500–4,499; 3,000 images) and reserve (4,500–4,999) sets. Development images were used to fix the methods and decision rules, the primary H1–H4 evaluation on the test images was run once after the protocol was locked, the later PAD16 comparison on the test images followed a separately locked supplementary plan, post hoc analyses reused existing records, and reserve images were used only for supplementary analyses (replication of the equal-capacity baseline, timing, image quality and the forgery experiment). For H1 and H4, five messages were drawn per image and all values of k = 1–5 and all 26 distortions were measured for the raw, BCH16, BCH21, RND16 and REP16 message sets. For H2, one scene was assigned to each image with fixed seeds (calibration 20261101, test 20261102): k uniform in 0–5 ($k = 0$ is the unwatermarked image), the distortion uniform over the 26 conditions, each message registered with probability 0.5, and unregistered messages either random or near-miss. Raw near-miss messages were constructed at Hamming distance 2 from a registered raw message, whereas BCH16 near-miss messages were codewords at the code's minimum distance of 8 from a registered codeword; the two settings therefore do not represent equal bit-distance perturbations. For H3, two messages were embedded per image: raw pairs at Hamming distance 1, 2, 3, 4, 6 or 8 or random, and BCH16 nearest pairs (distance 8) or random pairs, without distortion and under h-flip + contrast 1.5.

Metrics

The metrics are defined in the Results: exact message recovery by equation (1), the false-attribution and correct-attribution rates by equation (2), merging by equation (4) and the WAM metric and its overestimation by equation (5). FA concerns the acceptance of registered IDs that are absent from the scene; when several registered IDs are

embedded in one image, a region of one embedded user decoded as another embedded user is not counted in FA (it lowers R), so FA does not bound all region-to-user assignment errors. In all metrics a region is assigned to the message whose visible area it overlaps most, as in WAM, and undetected messages remain in the denominators.

Statistics and pre-specification

For a paired comparison, let d_i be the difference between two methods on image i (pairing the same image, k and region) and $\bar{d}$ its mean over the n images. The decision interval was

$$\left[\, q_{0.00625}\left(\bar{d}^{*}\right),\ q_{0.99375}\left(\bar{d}^{*}\right) \right], \tag{15}$$

where $\bar{d}^{*}$ are the means of 10,000 bootstrap[38] resamples of the images (seed 20261006) and q_p is the p-quantile, which gives a two-sided 98.75% interval (Bonferroni correction of 95% over four hypotheses). For x false attributions in n scenes, the one-sided 98.75% Clopper–Pearson upper bound[39] was

$$p_U = B^{-1}(0.9875;\ x+1,\ n-x), \tag{16}$$

where $B^{-1}(\cdot; a, b)$ is the quantile function of the beta distribution; for method C, $x = 3$ and $n = 3{,}000$ give the bound 0.32% of Table 1. Equation (15) produces the intervals of image-averaged paired differences in Table 1, Table 2 and Figs. 2–4, including the locked H2 statistic. For H2, let m_i be the number of registered eligible messages and $s_i^{(X)}$ the number of correct acceptances of method X in image i, and let I_+ be the n_+ images with $m_i > 0$; then

$$R_{\mathrm{msg}}(X) = \frac{\sum_{i\in I_+} s_i^{(X)}}{\sum_{i\in I_+} m_i}, \qquad R_{\mathrm{img}}(X) = \frac{1}{n_+}\sum_{i\in I_+} \frac{s_i^{(X)}}{m_i},$$
$$\Delta_{\mathrm{img}} = R_{\mathrm{img}}(C) - R_{\mathrm{img}}(A).$$

The R of equation (2) is R_{msg}, and the locked H2 statistic is Δ_{img}, whose interval follows equation (15) with $d_i = (s_i^{(C)} - s_i^{(A)})/m_i$. For the post hoc pooled differences $R_{\mathrm{msg}}(C) - R_{\mathrm{msg}}(A)$, images were resampled jointly across methods, and the pooled success and eligible-message counts were recomputed within each bootstrap replicate. Equation (16) produces every false-attribution upper bound. All conditions within a hypothesis had to hold (intersection–union test[40]), and equivalence was declared when the interval of the difference lay within the margin[42]. Secondary comparisons over distortions were Holm-corrected[41] and reported only; because the bootstrap p-value cannot fall below 1/10,001, the smallest Holm-adjusted value is reported as $p < 0.003$. The execution protocol (SHA-256 1b2a2d0fe9a0c7aa…) was locked at 2026-10-04T13:14:25Z together with the hashes of the checkpoints, code and per-split image lists, and the calibration and test runs verified all hashes before execution. Supplementary analyses—soft versus hard decoding, sensitivity to near-miss scores, the equal-capacity baseline on reserve and test images, timing, image quality and the forgery experiment—were each run under a plan locked by hash before measurement.

The primary analyses are the four locked hypothesis tests; the analyses above are supplementary. The following analyses were added after the locked analyses (post hoc) and reused existing records without new measurement: the stability of the calibrated threshold (2,000 resamples of the calibration scenes), the registry-size approximation, the merge differences at distances 3 and 4, the decomposition of failures in H4, and the pooled H2 differences with image-level bootstrap intervals. The locks are SHA-256 hashes that we recorded ourselves. They show that the locked files were not changed afterwards, but they do not independently certify when the files were created, because the plans were not deposited with an external registry before testing. All locked files and their recorded lock times are public in the code repository (Code availability), which was published after the analyses.

Forgery experiment

On the 500 reserve images, five registered target IDs per image were embedded in the checkerboard layout under public assignment (codeword index equal to ID) and under keyed assignment (codeword index $\pi_K(\mathrm{ID})$ for a secret uniform random permutation π_K of the 65,536 indices). The attacker, lacking the key, embeds the public codeword of the target. The supplementary decision rules were locked before measurement and compare region-level rates under an assumption of independent regions: keyed assignment was considered to have a lower targeted-forgery success rate than public assignment if the one-sided 98.75% upper bound of its region-level success rate was below the lower bound for public assignment (this was met), and to preserve legitimate attribution if the paired 98.75% interval of the difference lies within ± 0.02. All forgery rates are per region unless stated otherwise; the regions are clustered in images (five per image) and are therefore not independent. As an image-level event we count an image when at least one of its five forged regions is accepted as the target; with zero events in n images, the one-sided 98.75% upper bound is $1 - 0.0125^{1/n}$. These comparisons describe the tested attack and are not a general guarantee against impersonation. Attribution of forged regions to unrelated registered users was computed for the primary key and for 1,000 random keys from the same decoded indices, because decoding does not depend on the key; the key results therefore re-interpret one set of decoding records and are not 1,000 independent embedding and detection experiments. The reported rates are conditional on the primary key, and the distribution over keys describes how they vary with the key.

Reproducibility check

We re-implemented Algorithms 1 and 2 from the pseudocode, without the decoding and calibration functions of the study code. The reference implementation reproduced the decoded codeword of all 4,952 regions of the forgery experiment (4,952 identical, identical scores), the calibrated threshold ($\tau = 47.07576$), the test false attributions (3/3,000) and correct-attribution rate (0.80215), and the per-region acceptance counts (2,498/2,500 and 2,363/2,500).

Ethics declarations

This study did not involve human participants, human tissue or animals. All images are publicly available images of the COCO 2017 validation set[21], used under their Creative Commons licences; the image shown in Fig. 1 is licensed under CC BY 2.0 and was selected so that its captions mention no person. Ethical approval and informed consent were therefore not required.

References

[1] Cox, I. J., Kilian, J., Leighton, F. T. & Shamoon, T. Secure spread spectrum watermarking for multimedia. *IEEE Trans. Image Process.* **6**, pp. 1673–1687 (1997).

[2] Wen, Y., Kirchenbauer, J., Geiping, J. & Goldstein, T. Tree-rings watermarks: invisible fingerprints for diffusion images. In *Advances in Neural Information Processing Systems 36* pp. 58047–58063 (2023).

[3] Yu, N., Skripniuk, V., Abdelnabi, S. & Fritz, M. Artificial fingerprinting for generative models: rooting deepfake attribution in training data. In *Proc. IEEE/CVF International Conference on Computer Vision* pp. 14428–14437 (2021).

[4] Zhu, J., Kaplan, R., Johnson, J. & Fei-Fei, L. HiDDeN: hiding data with deep networks. In *Computer Vision – ECCV 2018* pp. 682–697 (Springer, 2018).

[5] Tancik, M., Mildenhall, B. & Ng, R. StegaStamp: invisible hyperlinks in physical photographs. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp. 2114–2123 (2020).

[6] Luo, X., Zhan, R., Chang, H., Yang, F. & Milanfar, P. Distortion agnostic deep watermarking. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp. 13545–13554 (2020).

[7] Jia, Z., Fang, H. & Zhang, W. MBRS: enhancing robustness of DNN-based watermarking by mini-batch of real and simulated JPEG compression. In *Proc. 29th ACM International Conference on Multimedia* pp. 41–49 (2021).

[8] Fernandez, P., Sablayrolles, A., Furon, T., Jégou, H. & Douze, M. Watermarking images in self-supervised latent spaces. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* pp. 3054–3058 (2022).

[9] Fernandez, P., Couairon, G., Jégou, H., Douze, M. & Furon, T. The Stable Signature: rooting watermarks in latent diffusion models. In *Proc. IEEE/CVF International Conference on Computer Vision* pp. 22409–22420 (2023).

[10] Bui, T., Agarwal, S. & Collomosse, J. TrustMark: universal watermarking for arbitrary resolution images. Preprint at https://arxiv.org/abs/2311.18297v1 (2023); cited content on PDF p. 8.

[11] Fernandez, P., Elsahar, H., Yalniz, I. Z. & Mourachko, A. Video Seal: open and efficient video watermarking. Preprint at https://arxiv.org/abs/2412.09492v1 (2024); cited content on PDF pp. 3–6 and 11.

[12] Sander, T., Fernandez, P., Durmus, A., Furon, T. & Douze, M. Watermark Anything with localized messages. In *International Conference on Learning Representations* (2025). Preprint at https://arxiv.org/abs/2411.07231v2; cited content on PDF pp. 4, 6–9, 23 and 26.

[13] Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd International Conference on Knowledge Discovery and Data Mining* pp. 226–231 (AAAI Press, 1996).

[14] Schubert, E., Sander, J., Ester, M., Kriegel, H.-P. & Xu, X. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Trans. Database Syst.* **42**, pp. 1–21 (2017).

[15] Bose, R. C. & Ray-Chaudhuri, D. K. On a class of error correcting binary group codes. *Inf. Control* **3**, pp. 68–79 (1960).

[16] Hocquenghem, A. Codes correcteurs d'erreurs. *Chiffres* **2**, pp. 147–156 (1959).

[17] Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J. & Wasserman, L. Distribution-free predictive inference for regression. *J. Am. Stat. Assoc.* **113**, pp. 1094–1111 (2018).

[18] Bates, S., Candès, E., Lei, L., Romano, Y. & Sesia, M. Testing for outliers with conformal p-values. *Ann. Stat.* **51**, pp. 149–178 (2023).

[19] Angelopoulos, A. N. & Bates, S. Conformal prediction: a gentle introduction. *Found. Trends Mach. Learn.* **16**, pp. 494–591 (2023).

[20] Nosek, B. A., Ebersole, C. R., DeHaven, A. C. & Mellor, D. T. The preregistration revolution. *Proc. Natl Acad. Sci. USA* **115**, pp. 2600–2606 (2018).

[21] Lin, T.-Y. et al. Microsoft COCO: common objects in context. In *Computer Vision – ECCV 2014* pp. 740–755 (Springer, 2014).

[22] Forney, G. D. Generalized minimum distance decoding. *IEEE Trans. Inf. Theory* **12**, pp. 125–131 (1966).

[23] Chase, D. Class of algorithms for decoding block codes with channel measurement information. *IEEE Trans. Inf. Theory* **18**, pp. 170–182 (1972).

[24] Zhang, X. et al. EditGuard: versatile image watermarking for tamper localization and copyright protection. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp. 11964–11974 (2024).

[25] Zhang, X. et al. OmniGuard: hybrid manipulation localization via augmented versatile deep image watermarking. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp. 3008–3018 (2025).

[26] Hu, R. et al. Mask image watermarking. In *Advances in Neural Information Processing Systems* (2025). Preprint at https://arxiv.org/abs/2504.12739v3; cited content on PDF pp. 8 and 22.

[27] Kim, J., Kim, H. & Kim, Y.-S. CORE-BREW: LLR-based soft decoding for robust multi-bit LLM watermarking. Preprint at https://arxiv.org/abs/2606.24163v2 (2026); cited content on PDF pp. 1–3 and 5.

[28] Park, G.-H., Lee, J.-H., Kwak, H.-Y. & Yun, D.-Y. WeaveMark: robust and scalable multi-bit LLM watermarking via coded payload spreading. Preprint at https://arxiv.org/abs/2609.02177v2 (2026); cited content on PDF pp. 1–2 and 4.

[29] Fernandez, P., Chaffin, A., Tit, K., Chappelier, V. & Furon, T. Three bricks to consolidate watermarks for large language models. In *Proc. IEEE International Workshop on Information Forensics and Security* pp. 1–6 (2023).

[30] Kim, C., Min, K., Patel, M., Cheng, S. & Yang, Y. WOUAF: weight modulation for user attribution and fingerprinting in text-to-image diffusion models. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp. 8974–8983 (2024).

[31] Yu, N., Davis, L. & Fritz, M. Attributing fake images to GANs: learning and analyzing GAN fingerprints. In *Proc. IEEE/CVF International Conference on Computer Vision* pp. 7555–7565 (2019).

[32] Xie, Y., Chen, X., Ren, Z. & Su, W. J. Watermark in the classroom: a conformal framework for adaptive AI usage detection. Preprint at https://arxiv.org/abs/2507.23113v1 (2025); cited content on PDF pp. 1 and 4.

[33] Qiao, T. et al. CertDW: towards certified dataset ownership verification via conformal calibration. Preprint at https://arxiv.org/abs/2506.13160v2 (2025), to appear in *IEEE Trans. Pattern Anal. Mach. Intell.*; cited content on PDF pp. 1–2.

[34] Tardos, G. Optimal probabilistic fingerprint codes. *J. ACM* **55**, pp. 1–24 (2008).

[35] Boneh, D. & Shaw, J. Collusion-secure fingerprinting for digital data. *IEEE Trans. Inf. Theory* **44**, pp. 1897–1905 (1998).

[36] Gloaguen, T., Staab, R., Vero, M. & Vechev, M. Every bit, everywhere, all at once: a binomial multibit LLM watermark. Preprint at https://arxiv.org/abs/2605.11653v2 (2026); cited content on PDF pp. 1–2 and 6.

[37] Kirillov, A., He, K., Girshick, R., Rother, C. & Dollár, P. Panoptic segmentation. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp. 9396–9405 (2019).

[38] Efron, B. Bootstrap methods: another look at the jackknife. *Ann. Stat.* **7**, pp. 1–26 (1979).

[39] Clopper, C. J. & Pearson, E. S. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika* **26**, pp. 404–413 (1934).

[40] Berger, R. L. & Hsu, J. C. Bioequivalence trials, intersection-union tests and equivalence confidence sets. *Stat. Sci.* **11**, pp. 283–319 (1996).

[41] Holm, S. A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* **6**, pp. 65–70 (1979).

[42] Schuirmann, D. J. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *J. Pharmacokinet. Biopharm.* **15**, pp. 657–680 (1987).

[43] Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**, pp. 600–612 (2004).

[44] Kutter, M., Voloshynovskiy, S. V. & Herrigel, A. Watermark copy attack. In *Proc. SPIE 3971, Security and Watermarking of Multimedia Contents II* pp. 371–380 (2000).

[45] Zhao, X. et al. Invisible image watermarks are provably removable using generative AI. In *Advances in Neural Information Processing Systems 37* pp. 8643–8672 (2024).

[46] An, B. et al. WAVES: benchmarking the robustness of image watermarks. In *Proc. 41st International Conference on Machine Learning*, PMLR **235**, pp. 1456–1492 (2024).

[47] Jiang, Z., Zhang, J. & Gong, N. Z. Evading watermark based detection of AI-generated content. In *Proc. 2023 ACM SIGSAC Conference on Computer and Communications Security* pp. 1168–1181 (2023).

[48] Chou, C.-H. & Li, Y.-C. A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile. *IEEE Trans. Circuits Syst. Video Technol.* **5**, pp. 467–476 (1995).

Acknowledgements. This research was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2024 (Project Name: Global Talent Training

Program for Copyright Management Technology in Game Contents, Project Number: RS-2024-00396709, Contribution Rate: 50%). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25492536). This study was separately supported by the 2026 Hongik University Innovation Support Program Fund. This work was supported by 2026 Hongik University Research Fund. Computing resources were provided by the School of Games, Hongik University.

Author contributions. H.G.Y. conceived the study, designed and conducted the experiments, analysed the results and wrote the manuscript. H.Y.K. supervised the study and revised the manuscript. All authors reviewed the manuscript.

Data availability. The COCO 2017 images are publicly available from the COCO website (https://cocodataset.org). The WAM code and checkpoints are publicly available from the authors of ref. [12]. The locked execution protocol and analysis plans, the summary results and the figure data generated in this study are available at https://github.com/Michael-John-73/ccmw (archived version: https://doi.org/10.5281/zenodo.23225977). The per-image records are available from the corresponding author on reasonable request.

Code availability. The scripts used for calibration, testing, supplementary analyses, figures and tables are available at https://github.com/Michael-John-73/ccmw (archived version: https://doi.org/10.5281/zenodo.23225977).

Additional information. Competing interests: The authors declare no competing interests.
Supplementary information accompanies this paper (condition-wise results, supplementary analyses and locked analysis plans).
Correspondence and requests for materials should be addressed to H.Y.K.

Figures and figure legends

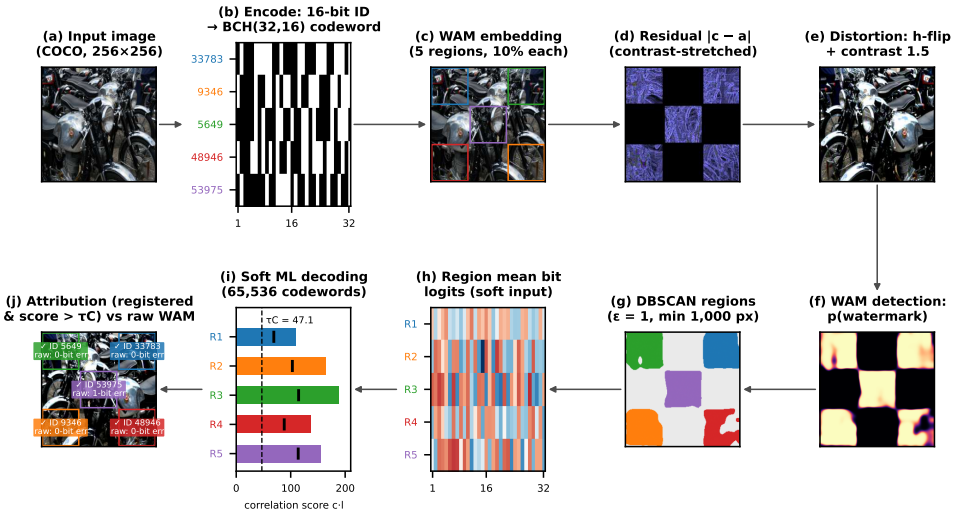


Fig. 1 Processing flow on one image. Codebook-constrained embedding (a–e) and soft decoding with calibrated attribution (f–j); the WAM model and its DBSCAN clustering are unchanged. The example is reserve COCO val2017 image 000000524108.jpg (Attribution License, CC BY 2.0). (a) Input resized to 256 × 256 pixels. (b) Five registered 16-bit IDs mapped to extended BCH(32,16) codewords (black, 1). (c) Each codeword embedded with the WAM embedder and pasted into a square of 10% area (WAM Sec. 5.5 layout, $k = 5$); peak signal-to-noise ratio (PSNR) 37.35 dB. (d) Absolute residual divided by its 99.5th percentile (0.043) for display. (e) Horizontal flip (h-flip) followed by contrast 1.5. (f) WAM per-pixel watermark probability. (g) WAM DBSCAN regions (Hamming $\varepsilon = 1$, minimum 1,000 pixels), coloured by the message they overlap most. (h) Mean bit logits of each region (equation (8)). (i) Correlation score of the best (bar) and second-best (tick) of the 65,536 codewords (soft maximum-likelihood, ML, decoding; equation (9)); dashed line, threshold $\tau_C = 47.08$ calibrated on 1,000 calibration images (equation (13)). (j) Accepted IDs (check marks) and, for the same image carrying random 32-bit messages, the number of bit errors of the raw WAM readout (DBSCAN centroid). The image is the first reserve image in name order (the 24th) with an open licence, no person mentioned in its captions, all five codewords accepted and at least one raw readout with 1–3 bit errors; it is illustrative only.

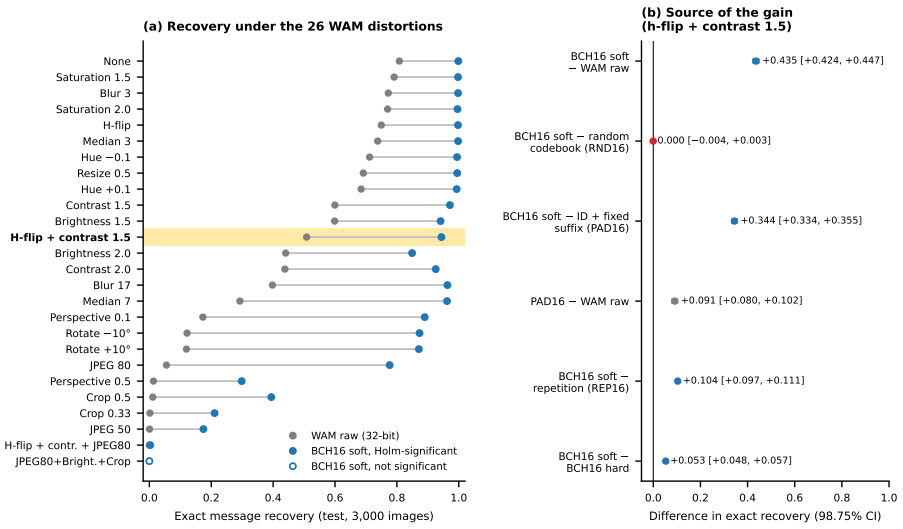


Fig. 2 Exact message recovery (H1). (a) Exact recovery (equation (1)) of WAM raw decoding (grey) and codebook-constrained soft decoding (BCH16, blue) for the 26 WAM evaluation distortions on the 3,000 test images ($k = 1$–5 messages pooled); filled circles, difference significant after Holm correction (25 of 26). Highlighted row: the decision condition (horizontal flip + contrast 1.5). (b) Paired differences with 98.75% bootstrap intervals in the decision condition: BCH16 versus WAM raw decoding, a random codebook of the same size (RND16), a 16-bit ID with a fixed 16-bit suffix (PAD16), a repetition code (REP16) and hard decoding of BCH16; PAD16 versus WAM raw decoding.

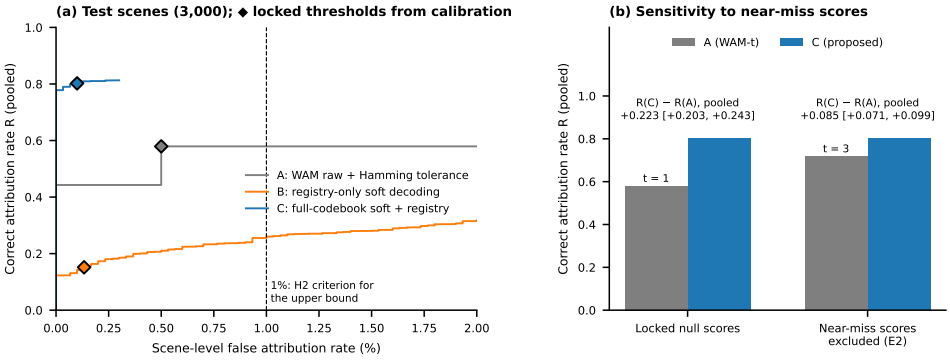


Fig. 3 Registry attribution (H2). (a) Scene-level false-attribution rate FA versus correct-attribution rate R (pooled, R_{msg}; equation (2)) on the 3,000 mixed test scenes as the acceptance threshold varies (descriptive); dashed line, 1% criterion of H2, which applies to the upper confidence bound rather than to the observed rate; diamonds, thresholds calibrated beforehand on 1,000 calibration images (Algorithm 2). A, WAM raw decoding with a Hamming tolerance; B, soft decoding within the registry only; C, full-codebook soft decoding with registry check. (b) Sensitivity analysis (E2): pooled correct attribution R at the calibrated thresholds with the locked null scores and with the null scores of near-miss messages (2-bit neighbours of registered raw messages) excluded before recalibration; the near-miss messages remain embedded. t, Hamming tolerance of A after calibration; differences are pooled, with 98.75% image-level bootstrap intervals (post hoc).

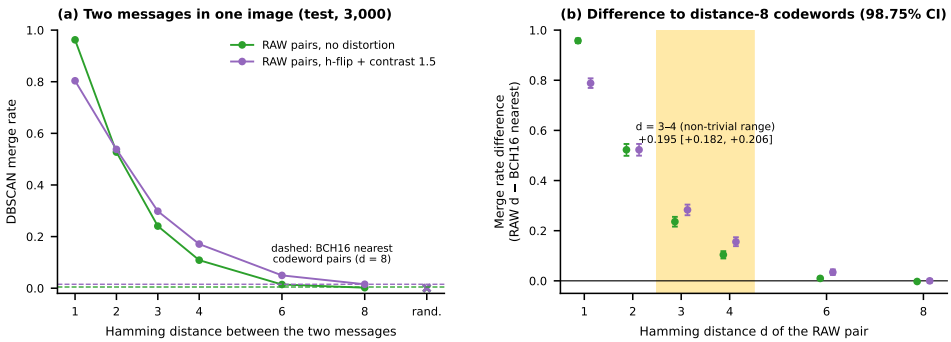


Fig. 4 Merging of nearby messages (H3). (a) DBSCAN merge rate (equation (4)) of two messages embedded in one image as a function of their Hamming distance (raw pairs), compared with nearest BCH16 codeword pairs (distance 8, dashed) and random pairs (crosses), without distortion and under horizontal flip + contrast 1.5 (3,000 test images). (b) Difference from the nearest BCH16 pairs with 98.75% bootstrap intervals (post hoc); distance 1 is expected from $\varepsilon = 1$, and the non-trivial range is distance 3–4 (shaded).

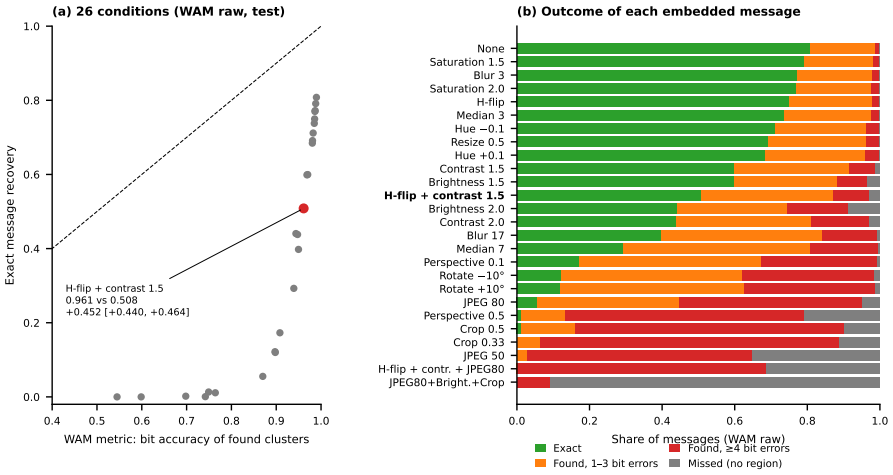


Fig. 5 Overestimation by the WAM metric (H4). (a) WAM's multi-message metric BA versus exact message recovery ER (equation (5)) for the 26 distortions (3,000 test images, WAM raw decoding); dashed line, equality. (b) Outcome of every embedded message under WAM raw decoding (post hoc): exact, found with 1–3 bit errors, found with 4 or more bit errors, or missed.

Tables

Table 1 Hypotheses, locked decision rules and test results. Test set: 3,000 COCO val2017 images; the primary evaluation was run once after the protocol was locked; checkpoint wam_coco. Intervals: 98.75% image-paired bootstrap (10,000 resamples; Bonferroni correction over four hypotheses). False-attribution bounds: one-sided 98.75% Clopper–Pearson. R is pooled over messages (R_{msg}); the H2 decision statistic Δ_{img} is the mean per-image difference in correct attribution (Methods). H3 (1) and H4 are expected from the definitions of DBSCAN ($\varepsilon = 1$) and of the WAM metric. The distance 3–4 difference (H3), the failure decomposition (H4) and the pooled H2 difference are post hoc analyses of the test records.

	Hypothesis	Locked decision rule	Test result	Verdict
H1	Codebook-constrained soft decoding recovers more messages exactly than WAM raw decoding.	Lower bound of BCH16 soft − WAM raw > 0 (h-flip + contrast 1.5, k = 1–5 pooled).	0.508 → 0.944; +0.435 [+0.424, +0.447].	Supported
H2	Calibrated full-codebook attribution (C) attributes more messages correctly than calibrated Hamming-tolerance attribution (A) at ≤ 1% false attribution.	Upper bounds on scene-level false attribution ≤ 1% for A and C; lower end of the 98.75% interval of Δ_{img} (C − A) > 0 (mixed scenes).	C 3/3,000 (≤ 0.32%), R_{msg} = 0.802; A 15/3,000 (≤ 0.87%), R_{msg} = 0.579; Δ_{img} = +0.213 [+0.193, +0.233] (R_{msg} difference +0.223 [+0.203, +0.243]).	Supported
H3	DBSCAN merges messages that are close in Hamming distance; codewords at distance 8 are rarely merged.	Lower bounds > 0 of (1) raw distance 1 − raw random and (2) raw distance ≤ 4 − nearest BCH16 pairs (merge rate).	(1) +0.883 [+0.872, +0.893]; (2) +0.446 [+0.437, +0.456]; distances 3–4: +0.195 [+0.182, +0.206].	Supported
H4	WAM's multi-message metric overestimates exact message recovery.	Lower bound of WAM metric − exact recovery > 0 (h-flip + contrast 1.5).	0.961 vs 0.508; +0.452 [+0.440, +0.464]; failures: 6.1% missed, 73.7% 1–3-bit, 20.2% ≥4-bit errors.	Supported

Table 2 Forgery by an attacker who knows the public codebook. Supplementary analysis on 500 reserve images (plan locked before measurement). The attacker has the same WAM embedder, the public codebook and decoding rule and the public ID of the target, but no key and no detector access. Public assignment: ID equals codeword index; keyed assignment: secret permutation of IDs to codewords. Method C with the calibrated threshold and the registry of 1,000 IDs. Rates are per forged region (five regions per image) unless stated otherwise; region-level upper bounds treat regions as independent, whereas the image-level row does not. Copying or replay, removal and adaptive attacks are not covered.

Measure	No distortion	H-flip + contrast 1.5
Legitimate attribution, public / keyed	0.999 / 0.998	0.945 / 0.948
Keyed − public (98.75% interval)	−0.002 [−0.004, 0.000]	+0.002 [−0.006, +0.011]
Targeted impersonation, public assignment	2,498/2,500	2,363/2,500
Targeted impersonation, keyed: successful regions (upper bound)	0/2,500 (≤ 0.18%)	0/2,500 (≤ 0.18%)
Targeted impersonation, keyed: images with at least one successful region (upper bound)	0/500 (≤ 0.87%)	0/500 (≤ 0.87%)
Attribution to unrelated registered users, keyed: primary key	1.64%	1.60%
median over 1,000 keys [5th, 95th percentile]	1.48% [0.84%, 2.36%]	1.44% [0.80%, 2.24%]
Expected value, $N/65{,}536 \times R_{\mathrm{legit}}$	1.52%	1.44%