# GEOMETRY-BASED WATERMARKING FOR VECTOR DATABASES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Vector databases are expensive to construct but can be readily replicated, creating a need for effective mechanisms to establish their provenance. Watermarking provides a potential solution; however, conventional coordinate-based watermarks can be disrupted by transformations that preserve Euclidean nearest-neighbor retrieval. To address this limitation, we propose a watermarking scheme that leverages local geometric structure both to identify watermark carriers and to encode the watermark message. In particular, distance ratios and angles provide invariant geometric fingerprints that remain stable under a broad class of transformations. Secret keys are then used to select watermark carriers based on these fingerprints, determine message positions, and select the angles used to encode individual bits. Verification requires only the suspect vector database and the corresponding secret key. We empirically evaluate the proposed scheme on four public datasets, with 50 trials conducted per dataset. Across a range of attacks, including multiple similarity transformations, vector shuffling, and index reconstruction, the method achieves a true-positive rate close to 100% at a 1% false-positive rate. Additional experiments evaluate the scheme's robustness to further forms of data modification and quantify the retrieval-quality loss incurred by attacks designed to compromise the watermark.

## 1 INTRODUCTION

Vector databases support retrieval-augmented generation, semantic search, and recommendation systems. They store documents, images, and other data as high-dimensional vectors and retrieve relevant items through nearest-neighbour search. These vectors form a valuable asset because they represent an organization's proprietary data in a searchable form. A vector database is expensive to produce and inexpensive to copy. Building one requires substantial computation to embed a large collection of data, yet a single export can transfer all the resulting vectors. The copied database remains useful for retrieval but provides no built-in way to verify its origin. Watermarking addresses this problem by embedding a signal determined by a secret key into the data (Petitcolas et al., 1999; Cox et al., 2008). The owner can then detect this signal in a suspect copy to verify its origin.

Two properties of vector data make watermarking difficult. First, applying the same rotation, reflection, translation, or uniform scaling to both the database and its queries leaves Euclidean nearest-neighbour results unchanged. These operations and their combinations are defined as *similarity transforms*. An adversary can therefore change the stored coordinates without affecting retrieval results (see e.g. the rotation experiment in Section 5). Second, writing a bit requires a small change to the vectors. The change must be large enough to read the bit later, but small enough to preserve retrieval quality. In the studied embeddings, a vector's nearest neighbours are often at nearly the same distance from it. This makes it difficult to write bits by changing these distances. A watermark must therefore remain readable under similarity transforms and work even when neighbour distances are nearly equal. The detector must also find and read the message using only the suspect database and the secret key. The main goal of this article is to develop such a watermarking scheme.

To that end, we introduce a new watermarking construction that combines invariant geometric fingerprints with keyed angular encoding. Specifically, local geometry identifies where the message is stored, and selected angles carry its bits. Then for each vector, we compute a fingerprint from distance ratios and angles involving its $k$ nearest neighbours. These fingerprints remain unchanged



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.

under similarity transforms. Using the fingerprints and a secret key, we select vectors to carry the message and assign a message bit to each one. The key also chooses two neighbours that form an angle at each selected vector. We store the bit by slightly moving one neighbour around that vector, changing the angle while keeping the distance between them fixed. Now, to read the message, we recompute the fingerprints and repeat the selection process on the suspect database. We then read the chosen angles and recover each message bit by majority vote across the vectors assigned to it.

The developed watermarking scheme was evaluated on four public datasets SIFT, DEEP, GIST, and nomic, with 50 independent trials per dataset. Under the tested similarity transforms, detection reaches a true-positive rate of near $100\%$ at a false-positive rate of $1\%$ on all four datasets. This performance is maintained when the vectors are shuffled and the search index is rebuilt. Embedding reduces recall@10, the fraction of a query's original ten nearest neighbours retained in the retrieved set, by less than $0.6$ percentage points on three datasets and $1.9$ on GIST. We further evaluated robustness to deletion, insertion, noise, and quantization, and measure the retrieval cost of an attack designed to disrupt the angles carrying the message.

Our main contributions are summarized below.

- **A watermark located and encoded through invariant geometry.** Local distance ratios and angles support carrier identification, keyed message positions, and angular encoding under similarity transforms. Detection uses only the suspect database and the secret key.
- **A coverage analysis for keyed message positions.** A bound relates the number of distinct carrier fingerprints to coverage of the message. It provides a reference for the anchor budget used in the experiments.
- **A comprehensive evaluation of robustness, retrieval quality, and scale.** Experiments compare four released baselines, evaluate 33 attack conditions, measure the cost of targeted removal, and test databases of up to 250,000 vectors on four datasets.

## 2 Related Work

**Watermarking vector databases.** Ren et al. (2025) introduce Random Selection (RS) and Transparent Vector Priority (TVP) for watermarking vector databases. Both derive identifiers from selected coordinate digits, partition vectors into message groups, and map each vector to a binary value. Coordinate bits are then modified to favour the target bit within each group, which is read by majority vote. RS selects vectors randomly. TVP prioritizes vectors expected to be retrieved less often, using HNSW edge lengths and degrees to estimate their query frequency. This reduces the effect of embedding on retrieval. Their evaluation covers deletion, insertion, noise, and an attack targeting transparent vectors. These methods make carrier selection sensitive to retrieval utility, but identification and bit recovery still depend on coordinate values. Our construction instead derives carrier identities, message positions, and encoding angles from local geometry, so the complete reading procedure is preserved under similarity transforms.

**Watermarking relational and tabular data.** Early relational watermarks modify low-order bits of numeric attributes (Agrawal & Kiernan, 2002). SCPW preserves statistical characteristics (Ren et al., 2024), while TabularMark uses controlled perturbations and hypothesis testing to protect datasets used for machine learning (Zheng et al., 2024). These approaches target the utility of tabular attributes. Vector databases additionally require robustness to coordinate transformations that preserve Euclidean neighbour rankings. The SCPW and TabularMark baselines follow the vector-data adaptations of Ren et al. (2025).

**Watermarking embedding services.** EmbMarker and WARDEN watermark embedding API outputs to detect model extraction (Peng et al., 2023; Shetty et al., 2024). GeoMark uses local regions of the embedding space to trigger a watermark associated with a shared target (Chen et al., 2026). These methods verify ownership through outputs produced by a suspect model. Our setting protects a stored vector collection and recovers its message directly from the collection and the owner's key.

**Geometric watermarking.** Geometric robustness has been studied for triangle meshes (Praun et al., 1999; Cho et al., 2007) and geographic vector data. Earlier map methods include reversible

watermarking of polygon coordinates (Voigt et al., 2004) and watermarking in singular values (Ren et al., 2021). Angles and distance ratios have also been used jointly for watermarking two-dimensional vector maps (Ren et al., 2022). Our contribution is a construction for unordered, high-dimensional vector databases. Local nearest-neighbour fingerprints determine both where a bit is stored and how its angle is read. This connects invariant identification, keyed message addressing, and angular encoding without requiring map connectivity or retained carrier locations. The evaluation measures both message recovery and the effect of embedding and attacks on retrieval.

## 3 Background and Problem Setting

**Database and retrieval.** Let $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ be a database. Euclidean distance is $d(x, y) = \|x - y\|_2$. A query $\xi \in \mathbb{R}^d$ returns the closest database vectors. For a stored vector $x$, let $y_1(x), \ldots, y_k(x)$ denote its $k$ nearest neighbours, excluding $x$ and ordered by increasing distance. Hereafter, these neighbours are written as $y_1, \ldots, y_k$ when the reference vector is clear. Write $d_i = d(x, y_i)$ for these distances. Retrieval quality is measured by recall@10, the average fraction of the original ten nearest neighbours retained in the returned set. Retrieval loss is the decrease from the original unwatermarked database; it includes the embedding loss when an attacked copy is evaluated.

**Similarity transforms.** A similarity transform has the form $T(x) = \lambda Rx + t$, where $R \in \mathbb{R}^{d\times d}$ is orthogonal, $\lambda > 0$ is a uniform scale factor, and $t \in \mathbb{R}^d$ is a translation vector. We denote the group of these transforms by $\mathcal{S}$. For any two vectors, $d(Tx, Ty) = \lambda\, d(x, y)$. All distances therefore change by the same factor, so their ordering is preserved. Applying the same $T$ to the database and its queries leaves nearest-neighbour results unchanged. Distance ratios and angles are also preserved.

**Watermarking objective.** The owner holds a secret key $\kappa$ and derives a binary message $W = (w_1, \ldots, w_N) \in \{0, 1\}^N$ from it using a pseudorandom function with a separate payload prefix. The database size is $n$; the message length is $N$. Embedding $W$ in $X$ produces $X'$ with little loss in retrieval quality. From a suspect database $Y$ and $\kappa$, the detector recovers a message $\widehat{W}(Y, \kappa)$ and tests its agreement with $W$. Neither the original database nor a stored list of modified vectors is required. The owner commits to the key before releasing $X'$ (Appendix C).

**Adversary model.** The adversary has the watermarked database $X'$ and knows the watermarking algorithm, but not the secret key $\kappa$. They may delete, insert, modify, or reorder vectors, rebuild the search index, and transform the database. Their goal is to remove the watermark while keeping the database useful for retrieval. We therefore evaluate both detection performance and the loss in retrieval quality caused by each attack. We assume that the adversary does not have the source data needed to regenerate the embeddings.

## 4 Method

Our method stores message bits in angles formed by nearby vectors. To find these angles again, we use fingerprints computed from local geometry and a secret key. Embedding consists of four steps. We choose a region of the database, compute fingerprints, use the key to select where each bit is stored, and adjust the selected angles. The fifth step describes detection, which repeats the selection process and reads the message from the suspect database. Figure 1 illustrates the process.

The owner derives a binary message $W = (w_1, \ldots, w_N) \in \{0, 1\}^N$ from the secret key $\kappa$. Here, $N$ is the message length. The detector uses the key to reconstruct the reference message $W$. It compares this reference with the message read from the suspect database. The reference message therefore does not need to be stored separately. Our goal is to embed these bits while preserving retrieval quality and allowing detection after an attack.

**Step 1. Choose the selection region.** We use the half of the database closest to its centroid as the *selection region*. All watermark modifications and the neighbourhood computations below are confined to this region. This keeps the message away from outlying vectors that an adversary might re-

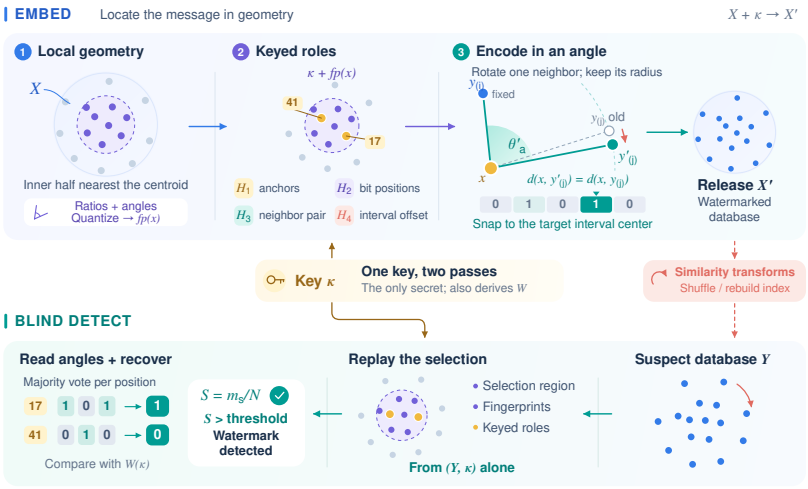


Figure 1: Watermark embedding and detection. **Embedding (top).** Within the selection region, geometric fingerprints and a secret key determine the anchors, message positions, and neighbour pairs. Each bit is written by moving one neighbour along an arc around its anchor, changing the selected angle while preserving their distance. **Detection (bottom).** From the suspect database and the key, the detector repeats the selection process, reads the angles, and combines votes to recover the message.

move. The region can be found again after any similarity transform, since these transforms preserve the ordering of distances to the centroid (Proposition 1, Appendix A). At detection, we recompute the centroid and the selection region from the suspect database.

**Step 2. Compute geometric fingerprints.** We describe each vector by its local geometry so that it can be recognized without relying on its coordinates. For each vector $x$ in the selection region, we find its $k$ nearest neighbours $y_1, \dots, y_k$ within that region. We compute

$$r_i = \log \frac{d(x, y_i)}{d(x, y_1)}, \qquad \phi_i = \angle(y_1, x, y_i), \qquad i = 2, \dots, k.$$

Here, $r_i$ is the log distance ratio relative to the closest neighbour. The angle $\phi_i$ is measured (in radians) at $x$, between the vectors $y_1 - x$ and $y_i - x$,

$$\phi_i = \arccos\left(\frac{(y_1 - x)^\top (y_i - x)}{\|y_1 - x\|_2 \, \|y_i - x\|_2}\right) \in [0, \pi].$$

We sort the distance and angle lists separately, from smallest to largest, and quantize them with step sizes $q_d$ and $q_\theta$. Concatenating the two lists gives the fingerprint

$$\mathrm{fp}(x) = \Big(Q_{q_d}(\mathrm{sort}_\uparrow(r_2, \dots, r_k)),\ Q_{q_\theta}(\mathrm{sort}_\uparrow(\phi_2, \dots, \phi_k))\Big),$$

where $Q_h(a) = \mathrm{round}(a/h)$ is applied to each entry. The function round returns the nearest integer. Rounding allows nearby feature values to receive the same integer code, so small changes within a bin leave the fingerprint unchanged. If its input is exactly halfway between two integers, it returns the even one. Sorting reorders the feature values within the fingerprint. The neighbours remain indexed by increasing distance from $x$. The fingerprint contains $2(k-1)$ integer entries and is unchanged under similarity transforms (Proposition 2, Appendix A). Also, small changes leave the fingerprint unchanged if no feature crosses a rounding boundary. Including angles helps distinguish vectors whose neighbour distances are nearly equal (Appendix E).

**Step 3. Select anchors and assign message positions.** We use the fingerprints and the secret key $\kappa$ to make three choices for each vector. First, we decide whether to select it as an anchor. For each selected anchor, we then choose a message position and the neighbour pair whose angle will

store the assigned bit. These choices are made by three keyed hash functions, $H_1$, $H_2$, and $H_3$, respectively. Each function returns a value in $[0, 1)$ and uses a distinct hash prefix (Appendix C).The key makes these choices reproducible for the owner without making them public. Separate hash prefixes keep anchor selection, message position, and neighbour-pair selection from using the same hash output.

*Anchor selection.* An anchor is the central vector of an angle used to store a message bit. To select $A$ anchors, we compute $H_1(\kappa, \mathrm{fp}(x))$ for every vector in the selection region and choose the $A$ lowest-scoring vectors. Let $\eta$ be the largest score among these selected vectors. Any additional vectors with score $\eta$ are also selected, so ties do not depend on storage order. Let $\mathcal{A}$ denote the set of selected anchors.

*Message-position assignment.* Each anchor $x$ is assigned a position in the message $W$ as,

$$\ell(x) = 1 + \lfloor N\, H_2(\kappa, \mathrm{fp}(x)) \rfloor \in \{1, \ldots, N\}.$$

For message position $i$, we group all anchors assigned to that position $\mathcal{A}_i = \{x \in \mathcal{A} : \ell(x) = i\}$. Each anchor in $\mathcal{A}_i$ is assigned the bit $w_i$. At detection, we read the bit from each selected anchor assigned to position $i$ and take a majority vote. A position may receive no votes, including when no anchor is assigned to it. If there are no votes, or the numbers of 0 and 1 votes are equal, the bit is left undecided (an erasure). Erasures count as errors in the detection score and BER (defined later).

*Angle selection.* We list the neighbour pairs in lexicographic order,

$$\mathcal{P} = \big((1,2), (1,3), \ldots, (1,k), (2,3), \ldots, (k-1,k)\big).$$

We number the pairs starting from zero. For example, pair 0 is $(1, 2)$ and pair 1 is $(1, 3)$. We write $\mathcal{P}[u]$ for the pair numbered $u$. The third hash selects the pair index,

$$p(x) = \left\lfloor \binom{k}{2} H_3(\kappa, \mathrm{fp}(x)) \right\rfloor, \qquad (i_x, j_x) = \mathcal{P}[p(x)].$$

Finally, the selected angle is $\theta_a(x) = \angle(y_{i_x}, x, y_{j_x})$. All three assignments can be recomputed from the database and the key.

**Theorem 1** (Message-position coverage)**.** *Suppose the selected anchors contain $A_{\mathrm{eff}}$ distinct fingerprints, chosen independently of $H_2$. Assume that $H_2$ assigns these fingerprints independently and uniformly to the $N$ message positions. Then*

$$\Pr[\textit{some message position is empty}] \le N(1 - 1/N)^{A_{\mathrm{eff}}} \le N e^{-A_{\mathrm{eff}}/N}.$$

*Hence, $A_{\mathrm{eff}} \ge N \ln(N/\varepsilon)$ is sufficient to cover all positions with probability at least $1 - \varepsilon$.*

The proof of Theorem 1 appears in Appendix A. For $N = 64$ and $\varepsilon = 0.01$, it suffices to have $A_{\mathrm{eff}} \ge 561$. Our experimental anchor budget is $A = 1122$, twice this count. Anchors with identical fingerprints share a message position and count only once toward $A_{\mathrm{eff}}$. The bound concerns coverage before embedding skips and read errors. Figure 5 reports the observed hash distributions.

**Step 4. Write the bit.** Step 3 assigns each anchor $x$ a message position $\ell(x)$ and a neighbour pair $(i_x, j_x)$. The bit to write is therefore $b_x = w_{\ell(x)}$, and its carrier is the angle $\theta_a(x)$. To store a bit, we move the selected angle to a nearby target value representing that bit. We use two sets of target angles, one for bit 0 and one for bit 1, following quantization-index modulation (Chen & Wornell, 2001). Specifically, targets for bit 0 are spaced $\Delta_\theta$ radians apart. Each target for bit 1 lies halfway between two targets for bit 0. This gives nearby choices for either bit, so we can write the assigned bit with a small angle change. At detection, we read the bit represented by the nearest target. A small change in the angle therefore does not change the bit unless it crosses the midpoint between two targets, assuming the same neighbour pair and target sets are recovered. We use $\Delta_\theta = 0.12$ radians for all anchors in the main experiments. Targets for the same bit are thus $0.12$ radians apart, and targets for opposite bits are separated by $0.06$ radians. Appendix D reports the sensitivity to this parameter. The procedure is detailed below.

The fourth keyed hash determines an offset for each anchor $\delta_x = \Delta_\theta H_4(\kappa, \mathrm{fp}(x))$. This keyed offset avoids using one public target grid for all anchors. It shifts the target angles without changing their

spacing. The detector recomputes the same offset from the key and fingerprint. The offset shifts all target angles by the same amount. It does not change their spacing. The target at $\delta_x$ represents bit 0. Moving along the angle axis in steps of $\Delta_\theta/2$, the labels alternate between 0 and 1,

| Target angle | $\delta_x$ | $\delta_x + \Delta_\theta/2$ | $\delta_x + \Delta_\theta$ | $\delta_x + 3\Delta_\theta/2$ |
|---|---|---|---|---|
| Bit | 0 | 1 | 0 | 1 |

The pattern extends in both directions, keeping only target angles in $[0, \pi]$. Thus, the targets for bits 0 and 1 are

$$\mathcal{C}_0(x) = \{\delta_x + m\Delta_\theta : m \in \mathbb{Z}\} \cap [0, \pi],$$
$$\mathcal{C}_1(x) = \{\delta_x + (m + \tfrac{1}{2})\Delta_\theta : m \in \mathbb{Z}\} \cap [0, \pi].$$

In other words, targets representing the same bit are $\Delta_\theta$ apart. To write $b_x$, we choose the nearest target angle labeled with that bit $\theta_a^*(x) = \arg\min_{u \in \mathcal{C}_{b_x}(x)} |u - \theta_a(x)|$. Thus, $b_x = 0$ selects a target from $\mathcal{C}_0(x)$, and $b_x = 1$ selects a target from $\mathcal{C}_1(x)$.

Note that we keep $x$ and $y_{i_x}$ fixed and move only $y_{j_x}$. Specifically, the neighbour $y_{j_x}$ moves along a circular arc centered at $x$, in the plane containing the three vectors. We stop when the angle reaches $\theta_a^*(x)$. Moving along this arc keeps its distance from $x$ unchanged. If $y'_{j_x}$ is the new position, then

$$\angle(y_{i_x}, x, y'_{j_x}) = \theta_a^*(x), \qquad d(x, y'_{j_x}) = d(x, y_{j_x}).$$

The detector reads the label of the nearest target angle. After the move, the angle is exactly at a target labeled $b_x$. The nearest target with the opposite label is $\Delta_\theta/2$ away. An angular change smaller than $\Delta_\theta/4$ therefore leaves the angle closer to the correct target. The bit is still read correctly, provided detection finds the same neighbour pair and the same offset.

All fingerprints and assignments are computed before embedding starts. We process anchors from the lowest selection score to the highest. After modifying an angle, we mark its three vectors $x$, $y_{i_x}$, and $y_{j_x}$ as used. Any later angle involving a used vector is skipped. We also skip cases where the rotation is undefined, such as coincident points. This prevents a vector from being used in more than one modified angle. Finally, multiple anchors are assigned to each message position when available. Their readings are combined by majority vote at detection. This redundancy helps tolerate skipped writes and fingerprint changes caused by embedding. Appendix C reports these effects.

**Step 5. Detect the watermark.** Given a suspect database $Y$ and the secret key $\kappa$, we repeat Steps 1–3 to compute the selection region, fingerprints, anchors, message positions, and neighbour pairs.

For each selected anchor $x$ with a defined angle, we recompute its offset from the key and its fingerprint in the suspect database $\delta_x = \Delta_\theta H_4(\kappa, \mathrm{fp}(x))$. We find the nearest target on the anchor's shifted grid and read its bit label. The formula below performs this calculation directly. Subtracting $\delta_x$ accounts for where the grid starts. Dividing by $\Delta_\theta/2$ and rounding gives the nearest target index. Even indices represent bit 0; odd indices represent bit 1.

$$\widehat{b}_x = \mathrm{round}\left(\frac{2(\theta_a(x) - \delta_x)}{\Delta_\theta}\right) \bmod 2.$$

This bit contributes one vote to message position $\ell(x)$.

Let $\mathcal{A}_i(Y)$ contain the selected anchors in $Y$ that are assigned to message position $i$ and have a defined angle $\theta_a(x)$. Each anchor contributes one vote for its recovered bit. The number of votes for $b \in \{0, 1\}$ is $V_i(b) = \sum_{x \in \mathcal{A}_i(Y)} \mathbf{1}\{\widehat{b}_x = b\}$. A strict majority determines the recovered bit,

$$\widehat{w}_i = \begin{cases} 0, & V_i(0) > V_i(1), \\ 1, & V_i(1) > V_i(0), \\ \perp, & V_i(0) = V_i(1). \end{cases}$$

Here, $\perp$ denotes an erasure. Positions with no votes or tied votes are erasures. The recovered message is $\widehat{W}(Y, \kappa) = (\widehat{w}_1, \ldots, \widehat{w}_N)$. The number of correctly recovered positions is $m_s = \sum_{i=1}^{N} \mathbf{1}\{\widehat{w}_i = w_i\}$. The detection score and bit error rate are

$$S(Y, \kappa) = \frac{m_s}{N}, \qquad \mathrm{BER} = 1 - S(Y, \kappa). \tag{1}$$

The score includes all $N$ message positions, with erasures counted as errors. We declare the watermark present when the score exceeds a threshold $\tau_\alpha$, i.e. $D(Y,\kappa) = \mathbf{1}\{S(Y,\kappa) > \tau_\alpha\}$. Here, $D(Y,\kappa) = 1$ means that the watermark is detected. We choose $\tau_\alpha$ from scores computed on unwatermarked databases. The threshold is set so that at most a fraction $\alpha$ of these calibration scores exceeds it. For $\alpha = 0.01$, this corresponds to the empirical 99th percentile. Scores equal to the threshold are not counted as detections, so ties can make the calibration FPR lower than 1%. An equivalent score is $z(Y,\kappa) = \frac{m_s - N/2}{\sqrt{N/4}} = 2\sqrt{N}\left(S(Y,\kappa) - \frac{1}{2}\right)$. Using the corresponding threshold $z_\alpha = 2\sqrt{N}\left(\tau_\alpha - \frac{1}{2}\right)$ gives exactly the same detection decision. Both thresholds come from the unwatermarked scores; no Gaussian approximation is used. The main evaluation uses 5,000 unwatermarked scores per dataset. The same scores are used to select the threshold and report the empirical FPR. Section 5.1 describes the sampling protocol. Appendix C provides the full embedding and detection algorithms.

# 5 Experiments

## 5.1 Setup

**Datasets.** The main experiments use 60,000 vectors per dataset: SIFT1M (128-d; Jégou et al., 2011), DEEP (96-d; Babenko & Lempitsky, 2016), GIST (960-d; Oliva & Torralba, 2001; Aumüller et al., 2020), and nomic arXiv embeddings (768-d; Nussbaum et al., 2025). DEEP and nomic are unit-normalized.

**Baselines.** We compare TVP and its random-selection variant RS (Ren et al., 2025), SCPW (Ren et al., 2024), and TabularMark (Zheng et al., 2024), using the implementations and adaptations in Ren et al. (2025). Baselines use native SIFT coordinates and fixed $\times 1000$ rescaling on the other datasets. Attacks use the same scaled coordinates, including their distance scale for relative noise. Ours, TVP, and RS use 64-bit messages; SCPW uses 32 bits. TabularMark reports its native tuple-matching score MP. Appendix F gives the full results.

**Configuration.** All four datasets use $k = 8$, $q_d = 0.08$, $q_\theta = 0.05$ radians, $N = 64$, $A = 1122$, and $\Delta_\theta = 0.12$ radians. Appendix D reports parameter and runtime measurements.

**Positive and negative samples.** Each of 50 trials watermarks a random 42,000-vector subset of the 60,000-vector pool using a fresh key. Each attack is applied to a separate copy, yielding 50 positive scores per condition. For negatives, 100 unwatermarked subsets of the same size are sampled separately from the original pool and evaluated under all 50 keys, yielding 5,000 scores. Positive and negative subsets may overlap. The negatives remain unattacked and are shared across attack conditions. The same negative scores calibrate the threshold and measure the empirical FPR.

**Metrics.** BER is reported as mean $\pm$ standard deviation over trials, with erasures counted as errors. Detection uses $S$ from Equation 1 and is summarized by AUROC and TPR at a 1% FPR. Retrieval cost is the recall@10 loss defined in Section 3, averaged over held-out queries and reported in percentage points. It includes embedding when an attacked database is evaluated. Full attack definitions appear in Appendix D.

## 5.2 Similarity transforms

After fixed rescaling where needed, all four baselines pass the clean detection check, but rotation disrupts their recovery (Table 1). Our method achieves AUROC 1.000 and 100% TPR at 1% FPR on all four datasets under every tested similarity transform, including combinations with shuffling and index reconstruction (Table 10, Appendix D). The methods offer complementary strengths: TVP and RS recover all bits under the tested pure deletion attacks, while our method preserves detection under similarity transforms. Appendix F reports the full comparison.

**Robustness to attacks.** Detection reaches 100% TPR at 1% FPR on all four datasets after 40% outlier deletion and 20% transparency deletion. Table 2 reports detection and retrieval cost under selected attacks; Appendix D covers all 33 conditions. PQ preserves recovery when detection reads

Table 1: Recovery before and after rotation over 50 trials. Entries are mean±standard deviation of BER (lower is better), except TabularMark, which reports MP (higher is better). SIFT uses native coordinates; the other datasets use fixed $\times 1000$ rescaling for the baselines. Ours, TVP, and RS use 64 bits; SCPW uses 32 bits. MP is not bit accuracy and can exceed one when identifiers collide.

| | SIFT | | DEEP | | nomic | | GIST | |
|---|---|---|---|---|---|---|---|---|
| | no-atk | rot | no-atk | rot | no-atk | rot | no-atk | rot |
| Ours | 0.007±0.010 | 0.007±0.010 | 0.016±0.015 | 0.016±0.015 | 0.021±0.017 | 0.021±0.016 | 0.130±0.036 | 0.133±0.037 |
| TVP | 0.000±0.000 | 0.512±0.059 | 0.000±0.000 | 0.493±0.063 | 0.000±0.000 | 0.499±0.073 | 0.000±0.000 | 0.505±0.063 |
| RS | 0.000±0.000 | 0.518±0.051 | 0.000±0.000 | 0.502±0.057 | 0.000±0.000 | 0.502±0.073 | 0.000±0.000 | 0.512±0.064 |
| SCPW | 0.000±0.000 | 0.508±0.074 | 0.000±0.000 | 0.480±0.083 | 0.000±0.000 | 0.484±0.090 | 0.000±0.000 | 0.524±0.076 |
| TabularMark (MP) | 1.000±0.000 | 0.000±0.000 | 1.000±0.000 | 0.000±0.000 | 1.000±0.000 | 0.000±0.000 | 1.010±0.000 | 0.000±0.000 |

Table 2: Detection and retrieval cost under selected attacks. AUROC and TPR use 50 positive scores and 5,000 unwatermarked negative scores per dataset. TPR is measured at 1% FPR. Cost is mean recall@10 loss in percentage points relative to the original database, including embedding. The same attack subset is shown for all four datasets. Full results appear in Tables 11–14.

| | SIFT | | | DEEP | | |
|---|---|---|---|---|---|---|
| Attack | AUROC | TPR | Cost | AUROC | TPR | Cost |
| Shuffle + rebuild + similarity | 1.000 | 1.00 | 0.3 | 1.000 | 1.00 | 0.4 |
| Outlier deletion, 40% | 1.000 | 1.00 | 36.6 | 1.000 | 1.00 | 34.6 |
| Random deletion, 10% | 1.000 | 1.00 | 10.1 | 1.000 | 1.00 | 10.2 |
| Random deletion, 20% | 0.939 | 0.50 | 20.0 | 0.936 | 0.50 | 20.0 |
| Gaussian noise, 0.161% | 1.000 | 1.00 | 1.4 | 1.000 | 1.00 | 1.8 |
| Transparency deletion, 20% | 1.000 | 1.00 | 9.8 | 1.000 | 1.00 | 9.2 |
| 8-bit quantization | 1.000 | 1.00 | 1.5 | 1.000 | 1.00 | 1.7 |
| 6-bit quantization | 0.769 | 0.08 | 4.8 | 0.555 | 0.02 | 5.6 |
| | nomic | | | GIST | | |
| Attack | AUROC | TPR | Cost | AUROC | TPR | Cost |
| Shuffle + rebuild + similarity | 1.000 | 1.00 | 0.6 | 1.000 | 1.00 | 1.9 |
| Outlier deletion, 40% | 1.000 | 1.00 | 33.0 | 1.000 | 1.00 | 19.4 |
| Random deletion, 10% | 1.000 | 1.00 | 10.2 | 0.997 | 0.94 | 10.9 |
| Random deletion, 20% | 0.958 | 0.30 | 20.1 | 0.812 | 0.32 | 20.4 |
| Gaussian noise, 0.161% | 1.000 | 0.98 | 2.3 | 0.908 | 0.26 | 4.4 |
| Transparency deletion, 20% | 1.000 | 1.00 | 8.8 | 1.000 | 1.00 | 4.2 |
| 8-bit quantization | 1.000 | 1.00 | 1.6 | 0.979 | 0.68 | 3.8 |
| 6-bit quantization | 0.805 | 0.12 | 4.5 | 0.520 | 0.02 | 10.5 |

retained original vectors, but reconstruction from PQ codes reduces recovery for our method, TVP, and RS (Tables 8 and 9).

**Utility and cost.** Embedding reduces recall@10 by less than $0.6$ percentage points on SIFT, DEEP, and nomic, and by $1.9$ on GIST (Table 7). Experiments also evaluate HNSW and scaling to 250,000 vectors on all four datasets. Appendix D reports runtime, message recovery, and retrieval loss for these settings.

**Sensitivity.** A sweep of 12 parameter configurations per dataset, with 50 trials each, examines neighbourhood size, fingerprint resolution, and encoding strength. Larger neighbourhoods reduce recovery under random deletion, while a coarser angle fingerprint step improves recovery under the tested noise and 8-bit quantization. Tables 20 and 21 report the full results. Appendix E reports neighbour-distance concentration and fingerprint distinctness.

**Security against an algorithm-aware adversary.** The adversary knows the algorithm but not the key and rotates neighbours around many centers to disrupt potential watermark angles. Figure 2 reports detection versus retrieval loss for four encoding strengths. At the default $\Delta_\theta = 0.12$, the first tested whole-database attack reaching AUROC $\leq 0.6$ incurs recall@10 losses of $8.6$–$20.2$ per-

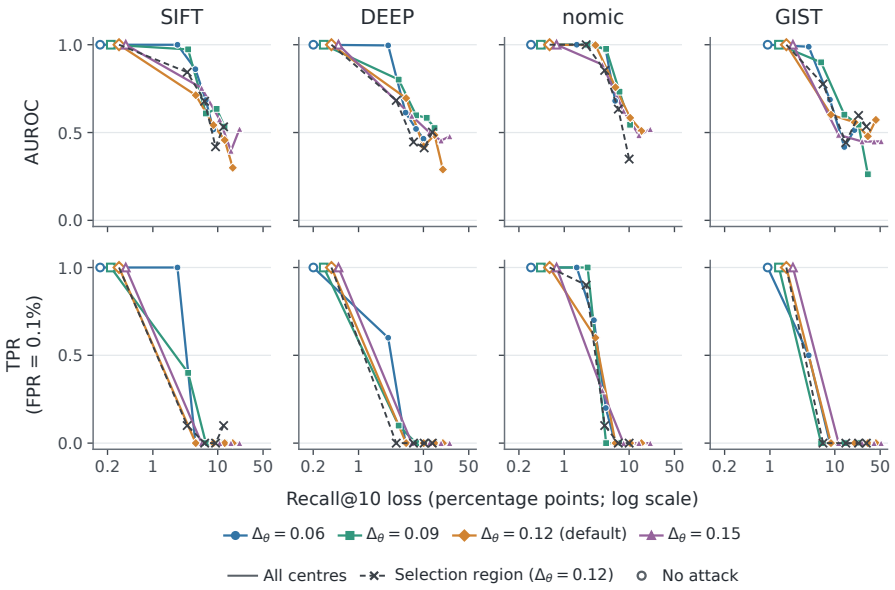


Figure 2: Detection versus recall@10 loss under the algorithm-aware angle attack. Columns show datasets; rows show AUROC and TPR at 0.1% FPR. Solid curves attack all centers; dashed curves attack the selection region at $\Delta_\theta = 0.12$. Open markers indicate embedding without attack. Loss includes embedding. Full settings appear in Appendix D.

centage points. Appendix D details the attacks on the whole database and on the selection region, with full numerical results.

## 6 Conclusion

We presented a watermark for vector databases that uses local geometry to both locate and encode the message. Distance ratios and angles make these operations invariant to similarity transforms. Detection requires only the suspect database and the secret key. Experiments on four public datasets demonstrate reliable detection under similarity transforms with little loss in retrieval quality. The method also tolerates substantial outlier deletion and several forms of noise and quantization. Our targeted angle attacks measure the retrieval cost of suppressing detection. Together, these results show that local geometry provides a practical basis for watermarking vector databases. Detection is less reliable under aggressive quantization and when only product-quantization reconstructed vectors are available. Future work will explore encoding schemes that better tolerate compression while preserving retrieval quality.

### AI use statement

Large language models were used only to assist with writing and polishing the manuscript. The authors reviewed all text and take full responsibility for the paper.

### Reproducibility statement

The appendices specify the embedding and detection algorithms, parameter settings, attack definitions, and evaluation protocols. The main experiments use 50 trial keys per dataset; the targeted attack sweep uses 10. Runtime tables distinguish timing repetitions from trials with new keys.

## REFERENCES


Rakesh Agrawal and Jerry Kiernan. Watermarking relational databases. In *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB)*, pp. 155–166. Morgan Kaufmann, 2002.

Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Information Systems*, 87:101374, 2020. doi: 10.1016/j.is.2019.02.006.

Artem Babenko and Victor Lempitsky. Efficient indexing of billion-scale datasets of deep descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2055–2063, 2016. doi: 10.1109/CVPR.2016.226.

Mihir Bellare. New proofs for NMAC and HMAC: Security without collision-resistance. In *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pp. 602–619. Springer, 2006. doi: 10.1007/11818175_36.

Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In *Advances in Cryptology – CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pp. 1–15. Springer, 1996. doi: 10.1007/3-540-68697-5_1.

Brian Chen and Gregory W. Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, 47(4):1423–1443, 2001. doi: 10.1109/18.923725.

Zhimin Chen, Xiaojie Liang, Wenbo Xu, Yuxuan Liu, and Wei Lu. Geometry-aware localized watermarking for copyright protection in embedding-as-a-service. *arXiv preprint arXiv:2604.11344*, 2026.

Jae-Won Cho, Rémy Prost, and Ho-Youl Jung. An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms. *IEEE Transactions on Signal Processing*, 55(1):142–155, 2007. doi: 10.1109/TSP.2006.882111.

Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2nd edition, 2008.

Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011. doi: 10.1109/TPAMI.2010.57.

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic Embed: Training a reproducible long context text embedder. *Transactions on Machine Learning Research*, 2025. URL `https://openreview.net/forum?id=IPmzyQSiQE`.

Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001. doi: 10.1023/A:1011139631724.

Wenjun Peng, Jingwei Yi, Fangzhao Wu, Shangxi Wu, Bin Zhu, Lingjuan Lyu, Binxing Jiao, Tong Xu, Guangzhong Sun, and Xing Xie. Are you copying my model? protecting the copyright of large language models for EaaS via backdoor watermark. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7653–7668. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.423.

Fabien A. P. Petitcolas, Ross J. Anderson, and Markus G. Kuhn. Information hiding—a survey. *Proceedings of the IEEE*, 87(7):1062–1078, 1999. doi: 10.1109/5.771065.

Emil Praun, Hugues Hoppe, and Adam Finkelstein. Robust mesh watermarking. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pp. 49–56. ACM, 1999. doi: 10.1145/311535.311540.

Na Ren, Ming Zhao, Changqing Zhu, Xiaohui Sun, and Yazhou Zhao. Commutative encryption and watermarking based on SVD for secure GIS vector data. *Earth Science Informatics*, 14(4): 2249–2263, 2021. doi: 10.1007/s12145-021-00684-5.

Na Ren, Deyu Tong, Hanchuan Cui, Changqing Zhu, and Qifei Zhou. Congruence and geometric feature-based commutative encryption-watermarking method for vector maps. *Computers & Geosciences*, 159:105009, 2022. doi: 10.1016/j.cageo.2021.105009.

Zhiwen Ren, Han Fang, Jie Zhang, Zehua Ma, Ronghao Lin, Weiming Zhang, and Nenghai Yu. A robust database watermarking scheme that preserves statistical characteristics. *IEEE Transactions on Knowledge and Data Engineering*, 36(6):2329–2342, 2024. doi: 10.1109/TKDE.2023.3324932.

Zhiwen Ren, Wei Fan, Qiyi Yao, Jing Qiu, Weiming Zhang, and Nenghai Yu. Vector database watermarking. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Anudeex Shetty, Yue Teng, Ke He, and Qiongkai Xu. WARDEN: Multi-directional backdoor watermarks for embedding-as-a-service copyright protection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13430–13444. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.725.

Michael Voigt, Bian Yang, and Christoph Busch. Reversible watermarking of 2D-vector data. In *Proceedings of the 2004 Workshop on Multimedia and Security (MM&Sec)*, pp. 160–165. ACM, 2004. doi: 10.1145/1022431.1022459.

Yihao Zheng, Haocheng Xia, Junyuan Pang, Jinfei Liu, Kui Ren, Lingyang Chu, Yang Cao, and Li Xiong. TabularMark: Watermarking tabular datasets for machine learning. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 3570–3584, 2024. doi: 10.1145/3658644.3690373.

# A PROOFS

The invariance results below concern similarity transforms in exact arithmetic. We assume nonzero neighbour distances and no ties at the selection-region boundary or in the neighbour rankings used by the method. With ties, the same results require a tie-breaking rule that is itself invariant to the transformation and to point reordering.

**Keyed functions and probability model.** HMAC-SHA256 is the keyed hash construction used for $H_1, \ldots, H_4$ and the payload generator (Bellare et al., 1996). Different prefixes separate their inputs. A pseudorandom function is a keyed function family whose outputs cannot be efficiently distinguished from those of a random function under its security assumptions (Bellare, 2006). The coverage analysis uses independent random functions as an ideal model of the separated mappings. The measured position histograms provide an empirical check of balance, not a proof of independence.

**Proposition 1** (Selection-region invariance)**.** *Let $T(x) = \lambda Rx + t$, where $\lambda > 0$ and $R$ is orthogonal. If $c$ is the centroid of $X$, then $Tc$ is the centroid of $TX$, and*

$$\|Tx - Tc\| = \lambda \|x - c\|.$$

*The ordering of distances to the centroid is therefore unchanged. Selecting the inner half of $TX$ returns the transformed selection region of $X$.*

*Proof.* The centroid transforms as

$$\frac{1}{|X|} \sum_{x \in X} (\lambda Rx + t) = \lambda Rc + t = Tc.$$

Orthogonality gives

$$\|Tx - Tc\| = \|\lambda R(x - c)\| = \lambda \|x - c\|.$$

Multiplication by the same positive constant preserves the ordering. □

**Proposition 2** (Fingerprint invariance)**.** *A similarity transform preserves the ordered neighbours used to construct a fingerprint. It also preserves their distance ratios and angles. Consequently,*

$$\mathrm{fp}(Tx) = \mathrm{fp}(x).$$

*Proof.* All distances are multiplied by $\lambda$, so neighbour order is unchanged and

$$\frac{\|Tx - Ty_i\|}{\|Tx - Ty_1\|} = \frac{\|x - y_i\|}{\|x - y_1\|}.$$

The logarithms of these ratios are therefore unchanged.

For $u = y_1 - x$ and $v = y_i - x$, orthogonality gives

$$\frac{\langle \lambda Ru, \lambda Rv \rangle}{\|\lambda Ru\| \|\lambda Rv\|} = \frac{\langle u, v \rangle}{\|u\| \|v\|}.$$

Thus the angles are unchanged as well. Applying the same quantization and sorting operations to identical values produces the same fingerprint. □

**Scope of the retrieval invariance.** For Euclidean retrieval, applying the same similarity transform to the database and queries preserves nearest-neighbour rankings. Cosine similarity is preserved by a common orthogonal transform and positive uniform scaling, but generally not by translation. The translation invariance claimed in this paper therefore concerns Euclidean retrieval.

**Selection under deletion with fixed scores.** If candidates are deleted while the surviving fingerprints and scores remain fixed, every surviving selected candidate stays selected. Fewer than $A$ candidates originally had a score strictly below its score, and deletion cannot increase that count. Ties at the cutoff are retained together. Recomputing the region or fingerprints can change these scores; those effects are measured separately.

**Message positions do not depend on row order.** The position $\ell(x) = 1 + \lfloor NH_2(\kappa, \mathrm{fp}(x)) \rfloor$ is determined by the key and fingerprint. It is unchanged whenever the fingerprint is preserved, regardless of insertions, deletions, or row permutations elsewhere in the database.

**Proof of Theorem 1.** A given position is missed by $A_{\mathrm{eff}}$ independent uniform assignments with probability $(1-1/N)^{A_{\mathrm{eff}}}$. The union bound over $N$ positions gives $N(1-1/N)^{A_{\mathrm{eff}}} \leq Ne^{-A_{\mathrm{eff}}/N}$. Requiring the last expression to be at most $\varepsilon$ yields $A_{\mathrm{eff}} \geq N \ln(N/\varepsilon)$. For $N = 64$ and $\varepsilon = 0.01$, rounding up gives 561.

The implementation maps a 53-bit hash output to one of 64 positions. This mapping is exactly uniform in the ideal random-function model, since 64 divides $2^{53}$. Identical fingerprints share an assignment, so the theorem counts distinct fingerprints rather than vectors. The budget $A = 1122 = 2 \times 561$ exceeds the sufficient count by a factor of two, but does not itself guarantee 561 distinct fingerprints. Default-setting counts are given in Table 3; the scaling results report how distinctness changes with database size.

## B Worked Example

This example follows one message carrier through fingerprinting, encoding, and detection. We use $k = 3$ and a small encoding step to make the geometry easy to inspect. Hash outputs are illustrative values, not outputs from an experimental key.

**Selection region.** Consider twelve points in the plane. Six central points are

$$\begin{aligned} p_1 &= (5.1, 4.8), \quad p_2 = (4.7, 5.3), \\ p_3 &= (5.9, 4.5), \quad p_4 = (4.2, 4.1), \\ p_5 &= (5.8, 5.9), \quad p_6 = (5.5, 5.2). \end{aligned}$$

The remaining points are

$$\begin{aligned} p_7 &= (9.8, 1.2), \qquad p_8 = (0.4, 9.1), \\ p_9 &= (9.5, 9.7), \qquad p_{10} = (0.2, 0.5), \\ p_{11} &= (10.4, 10.1), \quad p_{12} = (-0.6, 4.9). \end{aligned}$$

The centroid is approximately $(5.075, 5.442)$. The six points nearest to it are $p_1, \ldots, p_6$, which form the selection region.

**Fingerprint.** The three nearest in-region neighbours of $p_1$ are $p_6$, $p_2$, and $p_3$, at distances approximately $0.566$, $0.640$, and $0.854$. The two log-distance ratios are

$$\log \frac{d(p_1, p_2)}{d(p_1, p_6)} \approx 0.124, \qquad \log \frac{d(p_1, p_3)}{d(p_1, p_6)} \approx 0.412.$$

The angle features use the nearest neighbour $p_6$ as their common reference,

$$\angle(p_6, p_1, p_2) \approx 1.460, \qquad \angle(p_6, p_1, p_3) \approx 1.144,$$

with angles measured in radians. Quantizing these values and sorting each feature block gives $\mathrm{fp}(p_1)$.

**Keyed assignments.** Suppose the selection hash chooses $p_1$ as a carrier and

$$H_2(\kappa, \mathrm{fp}(p_1)) = 0.643.$$

Its message position is then

$$1 + \lfloor 64 \times 0.643 \rfloor = 42.$$

Suppose the pair hash selects $(p_6, p_2)$ and the dither is zero. These choices identify the angle in which the bit at position 42 will be written.

**Writing one bit.** For this illustration, use $\Delta_\theta = 0.02$ radians and target bit $w_{42} = 1$. The initial angle is approximately $1.46014$ radians. With zero dither, its decoded label is

$$\mathrm{round}\left(\frac{2 \times 1.46014}{0.02}\right) \bmod 2 = 146 \bmod 2 = 0.$$

Centers are $0.01$ radians apart, giving a decoding margin of $0.005$ radians at a center. The nearest center labeled 1 is $1.47$ radians. Rotating $p_2$ about $p_1$ increases the angle by approximately $0.00986$ radians and preserves $d(p_1, p_2)$. The illustrative encoding strength differs from the experimental value of $0.12$ radians.

**Reading the vote.** A similarity transform preserves the written angle of $1.47$ radians. If the carrier is reselected with the same fingerprint and pair, the key reconstructs position 42 and the zero dither. The vote is

$$\mathrm{round}\left(\frac{2 \times 1.47}{0.02}\right) \bmod 2 = 147 \bmod 2 = 1.$$

It contributes to position 42. The full detector combines all votes at that position and compares the recovered bit with $w_{42}$.

Similarity invariance applies to the geometry after embedding. Fingerprint changes caused by embedding itself, and their effect on carrier reselection, are described in Appendix C.

## C Implementation Details

**Fingerprint construction.** For each candidate, neighbours are ordered by distance within the selection region. The fingerprint contains $k - 1$ quantized log-distance ratios and $k - 1$ quantized angles. Each angle uses the nearest neighbour as its reference. The two blocks are sorted separately and concatenated. With $k = 8$, the fingerprint therefore contains 14 entries.

**Hash construction.** All keyed functions use HMAC-SHA256 (Bellare et al., 1996) with key $\kappa$. The quantized fingerprint is serialized into bytes. Separate prefixes distinguish the four functions,

$$\begin{aligned} H_1 &\quad \texttt{|}, \\ H_2 &\quad \texttt{slot|}, \\ H_3 &\quad \texttt{pair|}, \\ H_4 &\quad \texttt{dither|}. \end{aligned}$$

The leading 53 bits of each digest, divided by $2^{53}$, give a value in $[0, 1)$. The message $W$ is generated separately using the prefix `payload|` followed by a counter.

The message position and pair index are

$$\ell(x) = 1 + \lfloor N H_2(\kappa, \mathrm{fp}(x)) \rfloor, \qquad p(x) = \left\lfloor \binom{k}{2} H_3(\kappa, \mathrm{fp}(x)) \right\rfloor.$$

The paper numbers message positions from 1 to $N$; array indices in the implementation run from 0 to $N-1$. Pair indices start at zero and enumerate neighbour ranks $(1, 2), (1, 3), \ldots, (k-1, k)$ in lexicographic order. Thus $(i_x, j_x) = \mathcal{P}[p(x)]$, as in Step 3. The keyed offset is $\delta_x = \Delta_\theta H_4(\kappa, \mathrm{fp}(x))$. The decoder rounds $2(\theta_a(x) - \delta_x)/\Delta_\theta$ to the nearest integer and takes its parity. Ties use nearest-even rounding, as in the fingerprint quantizer. Detection recomputes neighbour ranks before applying the same mapping.

**Prior commitment to the key.** The ownership protocol assumes that the owner commits to the key before releasing the database, for example through a timestamped cryptographic hash. This fixes the key before the suspect database is examined. The reported false-positive rates concern such fixed keys; they do not account for searching over keys until a match is found.

Table 3: Embedding diagnostics at $\Delta_\theta = 0.12$. Results for $n = 60{,}000$ are from one run per dataset. Results for $n = 42{,}000$ are means over 50 trials; distinct-fingerprint counts are rounded. Fingerprint changes compare values before and after embedding. Candidate bit accuracy is measured by running detection on the watermarked database without an attack.

| | SIFT | DEEP | nomic | GIST |
|---|---|---|---|---|
| $n = 60{,}000$ | | | | |
| Distinct anchor fingerprints | 1120 | 1055 | 869 | 695 |
| Skipped candidates (%) | 10.8 | 13.6 | 13.3 | 27.8 |
| Candidate bit accuracy | 0.808 | 0.775 | 0.807 | 0.654 |
| Anchor angle block changed (%) | 28.5 | 32.1 | 28.7 | 45.3 |
| Anchor distance block changed (%) | 3.1 | 3.2 | 3.4 | 10.2 |
| Non-anchor fingerprint changed (%) | 9.9 | 13.5 | 12.2 | 34.9 |
| $n = 42{,}000$ | | | | |
| Distinct anchor fingerprints | 1121 | 1079 | 889 | 733 |
| Skipped candidates (%) | 15.2 | 18.7 | 19.1 | 33.6 |
| Candidate bit accuracy | 0.795 | 0.742 | 0.757 | 0.656 |

**Embedding order and skips.** Fingerprints, neighbour lists, and keyed assignments are computed once from the original database. Candidates are processed in ascending $H_1$ order.

A disjointness rule permits each vector to participate in at most one written triple, whether as the anchor or either neighbour. A candidate is skipped if its triple overlaps a previously written triple or its geometry does not support the required rotation.

The detector has no record of these skips. A skipped candidate can therefore be selected again and contribute a vote from an unmodified angle.

**Fingerprint stability after embedding.** Rotating a neighbour around its anchor preserves their distance. The anchor's distance features are unchanged by this individual operation, but its angle features can change. A write can also affect the fingerprints of other vectors whose neighbourhoods contain the moved point.

The implementation does not reject writes that cross fingerprint quantization boundaries. Table 3 reports the measured changes and the resulting candidate-level bit accuracy. This accuracy is the fraction of selected candidates with a finite carrier angle whose decoded bit matches the expected bit at their recomputed message position. It does not measure overlap with the original carrier set.

The selection region is also recomputed at detection. Across the measured configurations, at most 3 of the 1,122 original anchors leave the region after embedding.

**Message recovery and calibration.** The vote, erasure, score, and threshold definitions are given in Step 5 of Section 4. For the main experiment, 100 unwatermarked subsets of 42,000 vectors are each decoded under 50 keys, yielding 5,000 scores per dataset. The negatives are unattacked and reused across attack conditions. Threshold selection and the reported FPR use the same negative sample.

**Behaviour of the released baselines.** On SIFT, TVP, RS, and SCPW recover their messages before rotation. TabularMark returns its native tuple-matching score MP. These outputs allow each method's response to rotation to be measured using its own recovery metric.

At native coordinate scales on DEEP, nomic, and GIST, the released TVP and RS implementations exhibit identifier collapse. Their identifier construction first takes the absolute coordinate value, then extracts a digit using `str(int(value))[0]`. Coordinates with magnitude below one are truncated to zero by this operation, reducing the information available to distinguish vectors.

Native-scale failures are retained as implementation diagnostics. They are not used as the functional comparison in the main table. Fixed $\times 1000$ rescaling enables all four baselines on these datasets. TVP, RS, and SCPW then recover every bit in the clean trials. TabularMark also separates clean positives from negatives, but GIST retains minor identifier collisions: its clean MP is 1.01. MP is

Table 4: PCA alignment pilot without an attack. BER is reported as mean ± standard deviation over three trials per dataset and method. The retained-frame column uses the pre-embedding frame as a diagnostic control. The recomputed-frame column rebuilds the frame from the watermarked database.

| Dataset | Method | Retained-frame BER | Recomputed-frame BER |
|---|---|---|---|
| SIFT | TVP | $0.000 \pm 0.000$ | $0.490 \pm 0.100$ |
| | RS | $0.000 \pm 0.000$ | $0.448 \pm 0.130$ |
| DEEP | TVP | $0.000 \pm 0.000$ | $0.479 \pm 0.065$ |
| | RS | $0.000 \pm 0.000$ | $0.500 \pm 0.054$ |
| nomic | TVP | $0.000 \pm 0.000$ | $0.396 \pm 0.148$ |
| | RS | $0.000 \pm 0.000$ | $0.479 \pm 0.048$ |
| GIST | TVP | $0.000 \pm 0.000$ | $0.469 \pm 0.062$ |
| | RS | $0.000 \pm 0.000$ | $0.438 \pm 0.062$ |

**Algorithm 1** Embed$(X, \kappa)$

Generate the $N$-bit message $W$ from $\kappa$ using the payload prefix and counter
$C \leftarrow$ inner half of $X$ by distance to its centroid
Compute fingerprints and ordered in-region neighbour lists for all $x \in C$
Compute $H_1$ scores and select $\mathcal{A}$ using the $A$-th score cutoff
Precompute $\ell(x)$, $(i_x, j_x)$, and $\delta_x$ for every selected anchor
$X' \leftarrow X$; $U \leftarrow \varnothing$
**for** $x \in \mathcal{A}$ in ascending $H_1$ order **do**
  Retrieve $\ell(x)$, $y_{i_x}$, $y_{j_x}$, and $\delta_x$
  **if** $\{x, y_{i_x}, y_{j_x}\} \cap U = \varnothing$ and the geometry permits the rotation **then**
    $b_x \leftarrow w_{\ell(x)}$
    $\theta_a(x) \leftarrow \angle(y_{i_x}, x, y_{j_x})$ in $X'$
    $\theta_a^*(x) \leftarrow \arg\min_{u \in \mathcal{C}_{b_x}(x)} |u - \theta_a(x)|$
    Rotate $y_{j_x}$ about $x$ to reach $\theta_a^*(x)$, preserving $d(x, y_{j_x})$
    $U \leftarrow U \cup \{x, y_{i_x}, y_{j_x}\}$
  **end if**
**end for**
**return** $X'$

therefore retained as a native score, not converted to BER. DEEP and nomic are unit-normalized; GIST is not.

**PCA-based coordinate alignment.** A pilot tests whether PCA provides a reproducible coordinate frame for TVP and RS. Detection uses either the frame retained from before embedding or a frame recomputed from the watermarked database. No attack is applied. Both methods recover the message with the retained frame. Recomputing the frame gives BER between 0.396 and 0.500 (Table 4). The tested construction therefore fails the clean-recovery check, and the subsequent attack sweep is not run.

**Pseudocode.** The algorithms below use the fingerprint construction and keyed mappings defined above. Selecting the $A$ lowest scores includes the entire tied group at the cutoff. If fewer than $A$ candidates are available, all are selected. Embedding retains the initial assignments throughout the writing loop. Detection recomputes them from $Y$.

## D Attack Definitions and Extended Evaluation

This section describes the attacks, evaluation settings, and additional analyses. Tables 11, 12, 13, and 14 report the complete results.

**Algorithm 2** Detect$(Y, \kappa)$ at target FPR $\alpha$

Generate the same $N$-bit message $W$ from $\kappa$
$C \leftarrow$ inner half of $Y$ by distance to its centroid
Compute fingerprints and ordered in-region neighbour lists for all $x \in C$
Compute $H_1$ scores and select $\mathcal{A}$ using the $A$-th score cutoff
Set $V_i(0) = V_i(1) = 0$ for $i = 1, \ldots, N$
**for** $x \in \mathcal{A}$ **do**
  $\ell(x) \leftarrow 1 + \lfloor N H_2(\kappa, \mathrm{fp}(x)) \rfloor$
  $p(x) \leftarrow \lfloor \binom{k}{2} H_3(\kappa, \mathrm{fp}(x)) \rfloor$
  $(i_x, j_x) \leftarrow \mathcal{P}[p(x)]$
  $\delta_x \leftarrow \Delta_\theta H_4(\kappa, \mathrm{fp}(x))$
  $\theta_a(x) \leftarrow \angle(y_{i_x}, x, y_{j_x})$
  **if** $\theta_a(x)$ is finite **then**
    $\widehat{b}_x \leftarrow \mathrm{round}(2(\theta_a(x) - \delta_x)/\Delta_\theta) \bmod 2$
    $V_{\ell(x)}(\widehat{b}_x) \leftarrow V_{\ell(x)}(\widehat{b}_x) + 1$
  **end if**
**end for**
Recover $\widehat{W}$ by strict majority; empty or tied positions are erasures
$m_s \leftarrow \sum_{i=1}^{N} \mathbf{1}\{\widehat{w}_i = w_i\}$
$S \leftarrow m_s / N$
$D \leftarrow \mathbf{1}\{S > \tau_\alpha\}$, using the threshold from unwatermarked scores
**return** $\widehat{W}$, $S$, and $D$

**Default configuration.** Unless a sweep specifies otherwise, we use $k = 8$, $q_d = 0.08$, $q_\theta = 0.05$ radians, $\Delta_\theta = 0.12$ radians, $N = 64$, and $A = 1122$. The selection region contains the inner half of the database by distance to its centroid. The main evaluation uses 50 trials, each with a fresh key and a 42,000-vector subset of a 60,000-vector pool. Appendix C describes the negative samples and detection-score convention.

**Similarity transforms.** We test random orthogonal transforms generated by QR decomposition of Gaussian matrices, reflections, coordinate permutations, coordinate sign changes, translations, and uniform scaling by a factor of seven. The same transform is applied to database vectors and retrieval queries. The translated conditions in the final evaluation include an offset of $+100$.

**Deletion.** Extremal deletion removes the fraction $f_{\mathrm{del}}$ of vectors farthest from the database centroid. Random deletion removes a uniformly sampled fraction $f_{\mathrm{del}}$ of vectors. Transparency deletion follows Ren et al. (2025) and targets vectors preferred by their watermarking scheme.

These deletion patterns affect the watermark differently. Extremal deletion removes outer vectors first. Random deletion can remove carriers and their neighbours throughout the selection region, changing both the available votes and the fingerprints used to assign them.

**Insertion.** Random addition inserts vectors sampled from a separate unwatermarked pool. Adversarial addition inserts near-duplicates of existing vectors without access to the key. Each inserted vector is an existing point plus isotropic jitter scaled to $0.07$ times the dataset's median nearest-neighbour distance. Such insertions can enter existing neighbourhoods and change the fingerprints computed from them. The insertion fraction is measured relative to the database size before the attack.

**Noise and numerical representations.** The noise attacks add Gaussian or uniform perturbations at the strengths reported in the tables. The reported relative noise scale $\sigma_{\mathrm{rel}}$ is referenced to the median nearest-neighbour distance.

The quantization attacks round coordinates to 8-bit or 6-bit representations. The float16 attack casts coordinates to half precision and then restores the working representation. These operations perturb local distances and angles through numerical rounding.

Table 5: Retrieval cost of suppressing watermark detection under the whole-database angle attack. Each entry reports recall@10 loss in percentage points at the first tested attack strength where AUROC reaches 0.6 or below, using 10 trials per setting and no interpolation. Loss is measured relative to the original unwatermarked database and includes the embedding cost. NR indicates that the threshold was not reached; the parenthesized value is the largest tested loss.

| | $\Delta_\theta = 0.06$ | 0.09 | 0.12 | 0.15 |
|---|---|---|---|---|
| SIFT | 8.6 | 12.7 | 8.6 | 10.7 |
| DEEP | 7.7 | 7.7 | 10.1 | 6.6 |
| nomic | NR (6.1) | 10.3 | 10.4 | 14.2 |
| GIST | 14.0 | 23.2 | 20.2 | 11.5 |

**Other modifications.** The evaluation also includes a 2% shear, vector shuffling, index reconstruction, and the 30-coordinate modification attack adapted from Ren et al. (2025). Shuffling changes row order, and index reconstruction changes the retrieval index while retaining the stored vectors.

**Composite attacks.** C1 combines rotation, translation, and uniform scaling. C2 adds Gaussian noise to the similarity transform. C3 combines the similarity transform with random deletion. C4 combines it with extremal deletion. C5 combines the similarity transform, Gaussian noise, and random deletion. C6 combines vector shuffling, index reconstruction, and a similarity transform to model database redeployment.

**Algorithm-aware angle attack.** The attacker knows the method and its parameters, but not the secret key. For each attacked center, its $k$ neighbours are rotated around it by angles sampled uniformly from

$$(0, \rho\Delta_\theta), \qquad \rho \in \{0.25, 0.5, 0.75, 1\}.$$

Increasing $\rho$ increases the perturbation applied to local geometry.

The whole-database experiment uses

$$\Delta_\theta \in \{0.06, 0.09, 0.12, 0.15\}.$$

Each dataset–encoding-step–strength setting is evaluated over 10 trials with 42,000 vectors. Detection is compared against 1,000 unwatermarked negative scores per dataset and encoding step. This protocol is separate from the 50-trial evaluation of the main attack suite.

Figure 2 shows detection against retrieval loss. Table 5 reports the loss at the first tested strength where AUROC reaches 0.6 or below. No interpolation is used. A crossing does not imply that AUROC remains below the threshold at every stronger setting.

A restricted variant attacks only centers in the selection region at $\Delta_\theta = 0.12$. The region is recomputed from the published database without the key. Neighbours of these centers can lie outside the region and are still modified. Both scopes use the same trial keys, subsets, watermarked databases, and queries. Tables 18 and 19 report the complete paired comparison at the default encoding step.

At equal attack strength, the restricted variant has lower retrieval loss on all four datasets. At each scope's first tested crossing of AUROC $\leq 0.6$, its losses are 9.26, 7.04, 10.01, and 14.78 percentage points on SIFT, DEEP, nomic, and GIST. These costs are lower than whole-database attacks on three datasets; SIFT requires a higher strength and has a larger loss at the crossing.

**Encoding strength and attack cost.** The measured erasure cost need not increase monotonically with $\Delta_\theta$. For a fixed $\rho$, a larger encoding step also produces a larger attack rotation. The resulting curves therefore reflect changes in both embedding strength and attack magnitude.

On nomic at $\Delta_\theta = 0.06$, none of the tested strengths reaches the AUROC threshold. The corresponding table entry reports the largest tested retrieval loss as a lower bound within this attack sweep.

**Retrieval-cost measurement.** In the complete result tables, retrieval loss is measured relative to the original unwatermarked database. It includes the watermark's embedding cost. The no-attack rows therefore give the embedding losses reported in Table 7.

Table 6: Numerical-precision diagnostic on unwatermarked data. Entries are mean recall@10 losses in percentage points over 10 trials per dataset, relative to a float64 reference on the clean database. Original uses the float32 retrieval calculation. Re-centered subtracts a common center from the stored database and queries before float32 search. Float64 uses the transformed data and distance calculation in double precision. Embedding cost is not included.

| Dataset | Condition | Original | Re-centered | Float64 |
|---|---|---|---|---|
| SIFT | Translation | 0.0020 | 0.0015 | 0.0000 |
| | C1 | 0.0040 | 0.0030 | 0.0010 |
| | C5 | 10.4760 | 10.4760 | 10.4760 |
| DEEP | Translation | 85.1330 | 0.0055 | 0.0000 |
| | C1 | 4.4500 | 0.0015 | 0.0000 |
| | C5 | 85.7260 | 10.5695 | 10.5705 |
| nomic | Translation | 99.9710 | 0.0075 | 0.0000 |
| | C1 | 62.5515 | 0.0000 | 0.0000 |
| | C5 | 99.9705 | 10.6860 | 10.6850 |
| GIST | Translation | 99.9175 | 0.0485 | 0.0000 |
| | C1 | 43.0595 | 0.0455 | 0.0000 |
| | C5 | 99.9280 | 11.4150 | 11.4160 |

Queries are transformed with the database under similarity attacks. After deletion, surviving rows are mapped to their original identities when computing recall@10. The measurements use exact nearest-neighbour retrieval in float32.

**Translation and numerical precision.** A common translation preserves Euclidean distances. However, computing squared distances in float32 after a large translation can introduce numerical errors. These errors affect the retrieval costs marked † in Tables 12, 13, and 14.

A separate diagnostic evaluates translation, C1, and C5 on unwatermarked subsets over 10 trials per dataset. Table 6 compares the original float32 calculation with two alternatives. The first re-centers the stored database and queries before float32 search. The second evaluates the transformed data and distances in float64.

Re-centering reduces the large apparent losses under translation and C1 to near zero. C5 retains a retrieval cost from its noise and deletion components. These diagnostics exclude embedding and therefore do not replace the watermark-inclusive costs in the attack tables.

**Parameter sensitivity.** The final method is evaluated by varying $k$, $q_d$, $q_\theta$, and $\Delta_\theta$ one at a time. All other parameters remain at their default values. The sweep contains 12 distinct configurations per dataset, with 50 trials each, for 2,400 trials across four datasets.

Each configuration is tested without an attack and under rotation, random deletion of 10% and 20% of the vectors, Gaussian noise with relative scale 0.161%, and 8-bit coordinate quantization. Tables 20 and 21 report mean message-bit accuracy. All 64 message positions are included, with empty or tied positions counted as errors.

Table 20 varies the neighbourhood size. Table 21 varies fingerprint resolution and encoding strength. The default configuration is retained throughout the main evaluation.

Appendix E provides complementary measurements of local geometry and fingerprint distinctness.

**Retrieval utility and computational cost.** Table 7 reports retrieval loss over 50 trials. Table 15 gives a separate matched comparison of exact search and HNSW at the final configuration. Each backend uses the same data, key, message, and queries, with three timing repetitions on the same node. Both embedding and detection times include index construction and fingerprint computation. HNSW uses $M = 32$, $\mathrm{efConstruction} = 200$, and $\mathrm{efSearch} = 128$. The code uses 16 threads and Faiss 1.10.0.

Table 7: Retrieval quality after watermark embedding. Recall@10 loss is reported in percentage points relative to the unwatermarked database, as mean ± standard deviation over 50 trials with 42,000 vectors each.

| | SIFT | DEEP | nomic | GIST |
|---|---|---|---|---|
| recall@10 loss | 0.32±0.03 | 0.40±0.04 | 0.59±0.06 | 1.85±0.13 |

The recovered message is identical across backends on SIFT, DEEP, and nomic in these measurements. On GIST, the recovered vectors of bits and erasures differ at eight positions, and BER changes from 0.09375 to 0.125. Retrieval is evaluated with exact search for both backends; different watermark modifications can still change its result. The GIST recall loss is 1.800 and 1.845 percentage points, respectively.

**Scaling with database size.** Tables 16 and 17 report the final configuration on all four datasets, from 15,000 to 250,000 genuine vectors. Database prefixes are nested; evaluation queries and the calibration pool are disjoint from the database. The target anchor budget remains $A = 1122$ as the database grows. The number of completed writes can change because overlapping triples are skipped. This fixed budget limits the fraction of vectors modified at large database sizes. Distinct-fingerprint counts and BER are reported alongside timing, since a fixed budget alone does not guarantee message recovery at every size.

**Product quantization.** Product quantization (PQ) is evaluated in two storage settings. In the first, the original watermarked vectors are retained alongside the PQ index, and detection reads those vectors. In the second, only PQ codes remain, and detection reads the reconstructed vectors.

Each subquantizer encodes a block of $d_{\mathrm{sub}}$ coordinates using 8 bits and 256 codewords. The tested block sizes are 4, 8, and 16. Table 8 reports results when the quantizer is trained on a separate unwatermarked pool. Each setting uses 10 trials. Our method uses a 64-bit message; TVP and RS use 32-bit messages in this experiment.

Keeping the original vectors preserves message recovery. Recovery degrades when only reconstructed vectors are available. The table also reports the retrieval loss of the PQ index to show the cost associated with each compression setting.

The second experiment trains the quantizer directly on the copied watermarked database. All other settings remain unchanged. Table 9 reports the results using the same storage settings and metrics as Table 8.

**Conditions that reduce detection.** Detection becomes less reliable under 6-bit quantization, 20% random deletion, and combinations of geometric transforms, noise, and deletion. GIST is also more sensitive to noise and 8-bit quantization than the other datasets. Appendix D analyses these cases and their effects on retrieval quality.

**Detection across attacks.** Figure 3 summarizes detection performance across the attack suite. Figure 4 summarizes message accuracy across trials, together with the unwatermarked reference scores.

**Geometric features and message addressing.** Table 22 summarizes the spread of carrier angles and nearest-neighbour distance ratios. Appendix E explains their implications for encoding. Figure 5 examines $H_2$ values over distinct fingerprints and complements Theorem 1.

Table 8: PQ trained on a separate unwatermarked pool. Original and Reconstructed report mean bit accuracy when detection reads the retained original vectors or their PQ reconstructions, respectively. AUROC and TPR refer to detection from reconstructions. TPR uses a nominal 1% FPR, calibrated on 100 PQ-processed unwatermarked scores per setting. The same negatives are used for calibration and evaluation. Loss is mean recall@10 loss of the PQ index, in percentage points relative to the clean database. All settings use 10 trials.

| Dataset | $d_{\mathrm{sub}}$ | Method | Original | Reconstructed | AUROC | TPR | Loss |
|---|---|---|---|---|---|---|---|
| SIFT | 4 | Ours | 0.997 | 0.480 | 0.605 | 0.10 | 24.23 |
| | 4 | TVP | 1.000 | 0.503 | 0.526 | 0.00 | 24.98 |
| | 4 | RS | 1.000 | 0.506 | 0.540 | 0.00 | 24.98 |
| | 8 | Ours | 0.997 | 0.423 | 0.331 | 0.00 | 38.77 |
| | 8 | TVP | 1.000 | 0.525 | 0.489 | 0.00 | 39.86 |
| | 8 | RS | 1.000 | 0.528 | 0.504 | 0.00 | 39.86 |
| | 16 | Ours | 0.997 | 0.455 | 0.523 | 0.00 | 57.70 |
| | 16 | TVP | 1.000 | 0.491 | 0.473 | 0.00 | 59.34 |
| | 16 | RS | 1.000 | 0.494 | 0.491 | 0.00 | 59.35 |
| DEEP | 4 | Ours | 0.977 | 0.458 | 0.512 | 0.00 | 30.77 |
| | 4 | TVP | 1.000 | 0.512 | 0.439 | 0.00 | 32.69 |
| | 4 | RS | 1.000 | 0.519 | 0.441 | 0.10 | 33.16 |
| | 8 | Ours | 0.977 | 0.434 | 0.389 | 0.00 | 53.19 |
| | 8 | TVP | 1.000 | 0.537 | 0.432 | 0.00 | 55.71 |
| | 8 | RS | 1.000 | 0.544 | 0.475 | 0.00 | 55.99 |
| | 16 | Ours | 0.977 | 0.469 | 0.608 | 0.00 | 72.90 |
| | 16 | TVP | 1.000 | 0.494 | 0.400 | 0.00 | 75.41 |
| | 16 | RS | 1.000 | 0.494 | 0.391 | 0.00 | 75.58 |
| nomic | 4 | Ours | 0.986 | 0.461 | 0.564 | 0.00 | 14.81 |
| | 4 | TVP | 1.000 | 0.441 | 0.414 | 0.00 | 14.90 |
| | 4 | RS | 1.000 | 0.450 | 0.438 | 0.00 | 14.99 |
| | 8 | Ours | 0.986 | 0.461 | 0.638 | 0.00 | 28.71 |
| | 8 | TVP | 1.000 | 0.516 | 0.521 | 0.00 | 29.12 |
| | 8 | RS | 1.000 | 0.500 | 0.466 | 0.00 | 29.17 |
| | 16 | Ours | 0.986 | 0.416 | 0.348 | 0.00 | 45.56 |
| | 16 | TVP | 1.000 | 0.466 | 0.433 | 0.00 | 46.26 |
| | 16 | RS | 1.000 | 0.475 | 0.463 | 0.00 | 46.32 |
| GIST | 4 | Ours | 0.856 | 0.467 | 0.574 | 0.10 | 20.90 |
| | 4 | TVP | 1.000 | 0.522 | 0.542 | 0.00 | 21.37 |
| | 4 | RS | 1.000 | 0.519 | 0.530 | 0.00 | 21.37 |
| | 8 | Ours | 0.856 | 0.412 | 0.365 | 0.00 | 40.30 |
| | 8 | TVP | 1.000 | 0.456 | 0.442 | 0.00 | 41.82 |
| | 8 | RS | 1.000 | 0.456 | 0.442 | 0.00 | 41.82 |
| | 16 | Ours | 0.859 | 0.448 | 0.439 | 0.00 | 58.03 |
| | 16 | TVP | 1.000 | 0.469 | 0.506 | 0.00 | 59.64 |
| | 16 | RS | 1.000 | 0.472 | 0.529 | 0.00 | 59.64 |

Table 9: PQ trained on the copied watermarked database. Original and Reconstructed report mean bit accuracy when detection reads the retained original vectors or their PQ reconstructions, respectively. AUROC and TPR refer to detection from reconstructions. TPR uses a nominal 1% FPR, calibrated on 100 PQ-processed unwatermarked scores per setting. The same negatives are used for calibration and evaluation. Loss is mean recall@10 loss of the PQ index, in percentage points relative to the clean database. All settings use 10 trials. Our method uses 64-bit messages; TVP and RS use 32-bit messages.

| Dataset | $d_{\mathrm{sub}}$ | Method | Original | Reconstructed | AUROC | TPR | Loss |
|---|---|---|---|---|---|---|---|
| SIFT | 4 | Ours | 0.997 | 0.455 | 0.524 | 0.00 | 23.43 |
| | 4 | TVP | 1.000 | 0.541 | 0.494 | 0.00 | 24.37 |
| | 4 | RS | 1.000 | 0.463 | 0.463 | 0.00 | 24.32 |
| | 8 | Ours | 0.997 | 0.481 | 0.641 | 0.20 | 36.76 |
| | 8 | TVP | 1.000 | 0.509 | 0.337 | 0.00 | 38.31 |
| | 8 | RS | 1.000 | 0.500 | 0.488 | 0.00 | 38.26 |
| | 16 | Ours | 0.997 | 0.423 | 0.364 | 0.00 | 56.08 |
| | 16 | TVP | 1.000 | 0.531 | 0.559 | 0.00 | 57.75 |
| | 16 | RS | 1.000 | 0.481 | 0.450 | 0.00 | 57.74 |
| DEEP | 4 | Ours | 0.977 | 0.469 | 0.589 | 0.00 | 29.90 |
| | 4 | TVP | 1.000 | 0.534 | 0.541 | 0.00 | 32.02 |
| | 4 | RS | 1.000 | 0.456 | 0.395 | 0.00 | 32.59 |
| | 8 | Ours | 0.977 | 0.439 | 0.411 | 0.00 | 52.03 |
| | 8 | TVP | 1.000 | 0.472 | 0.416 | 0.00 | 54.64 |
| | 8 | RS | 1.000 | 0.503 | 0.502 | 0.00 | 55.04 |
| | 16 | Ours | 0.977 | 0.448 | 0.489 | 0.00 | 71.83 |
| | 16 | TVP | 1.000 | 0.494 | 0.577 | 0.00 | 74.62 |
| | 16 | RS | 1.000 | 0.537 | 0.571 | 0.00 | 74.66 |
| nomic | 4 | Ours | 0.986 | 0.434 | 0.425 | 0.00 | 14.39 |
| | 4 | TVP | 1.000 | 0.509 | 0.557 | 0.00 | 14.65 |
| | 4 | RS | 1.000 | 0.519 | 0.534 | 0.10 | 14.79 |
| | 8 | Ours | 0.986 | 0.453 | 0.594 | 0.00 | 27.98 |
| | 8 | TVP | 1.000 | 0.516 | 0.514 | 0.00 | 28.32 |
| | 8 | RS | 1.000 | 0.591 | 0.640 | 0.10 | 28.46 |
| | 16 | Ours | 0.986 | 0.436 | 0.512 | 0.00 | 44.33 |
| | 16 | TVP | 1.000 | 0.522 | 0.601 | 0.00 | 45.35 |
| | 16 | RS | 1.000 | 0.547 | 0.580 | 0.00 | 45.37 |
| GIST | 4 | Ours | 0.856 | 0.467 | 0.610 | 0.00 | 20.15 |
| | 4 | TVP | 1.000 | 0.522 | 0.580 | 0.00 | 20.94 |
| | 4 | RS | 1.000 | 0.475 | 0.423 | 0.00 | 20.90 |
| | 8 | Ours | 0.856 | 0.412 | 0.326 | 0.00 | 39.00 |
| | 8 | TVP | 1.000 | 0.478 | 0.539 | 0.00 | 40.78 |
| | 8 | RS | 1.000 | 0.512 | 0.526 | 0.00 | 40.76 |
| | 16 | Ours | 0.859 | 0.452 | 0.498 | 0.00 | 56.17 |
| | 16 | TVP | 1.000 | 0.506 | 0.574 | 0.00 | 58.14 |
| | 16 | RS | 1.000 | 0.475 | 0.479 | 0.00 | 58.16 |

Table 10: Watermark detection before and after rotation. For each dataset, bit-accuracy scores from 50 watermarked instances are compared with 5,000 scores from unwatermarked subsamples decoded under the same keys. The table reports mean positive and negative scores, the largest negative score, AUROC, and TPR at the indicated false-positive rates. TVP is shown on SIFT, where it recovers the message before rotation.

| dataset | condition | pos. acc | neg. acc | neg. max | AUROC | $TPR_{1\%}$ | $TPR_{0.1\%}$ |
|---|---|---|---|---|---|---|---|
| SIFT | no attack | 0.993 | 0.453 | 0.719 | **1.000** | **1.00** | **1.00** |
| SIFT | rotation | 0.993 | 0.453 | 0.719 | **1.000** | **1.00** | **1.00** |
| DEEP | no attack | 0.984 | 0.452 | 0.719 | **1.000** | **1.00** | **1.00** |
| DEEP | rotation | 0.984 | 0.452 | 0.719 | **1.000** | **1.00** | **1.00** |
| nomic | no attack | 0.979 | 0.454 | 0.688 | **1.000** | **1.00** | **1.00** |
| nomic | rotation | 0.979 | 0.454 | 0.688 | **1.000** | **1.00** | **1.00** |
| GIST | no attack | 0.870 | 0.451 | 0.719 | **1.000** | **1.00** | **1.00** |
| GIST | rotation | 0.867 | 0.451 | 0.719 | **1.000** | **1.00** | **1.00** |
| SIFT (TVP) | no attack | 1.000 | 0.495 | 0.656 | 1.000 | 1.00 | 1.00 |
| SIFT (TVP) | rotation | 0.487 | 0.495 | 0.656 | 0.459 | 0.00 | 0.00 |

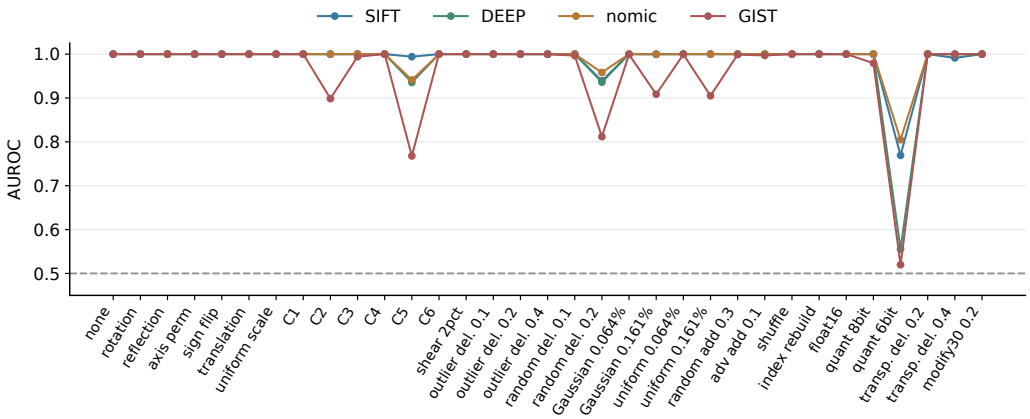


Figure 3: Detection performance across attack conditions. AUROC compares 50 watermarked trials with 5,000 unwatermarked detection scores per dataset. The same negative scores are used across conditions. The dashed line marks chance-level AUROC.

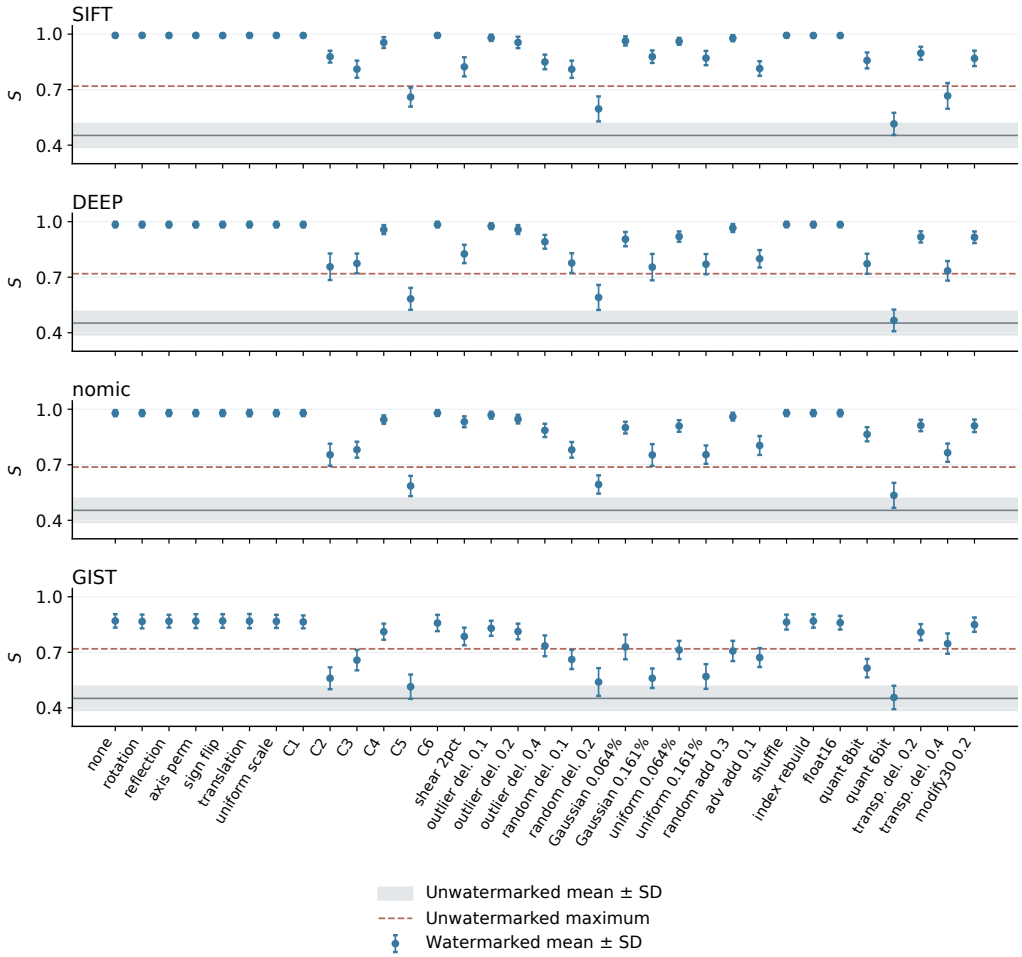


Figure 4: Message-score summaries across the 33 main conditions. Blue points and error bars show mean ± standard deviation over 50 watermarked trials. The grey band shows the mean ± standard deviation of 5,000 unwatermarked scores, with their maximum marked by the dashed line. All scores use all 64 message positions; erasures count as errors.

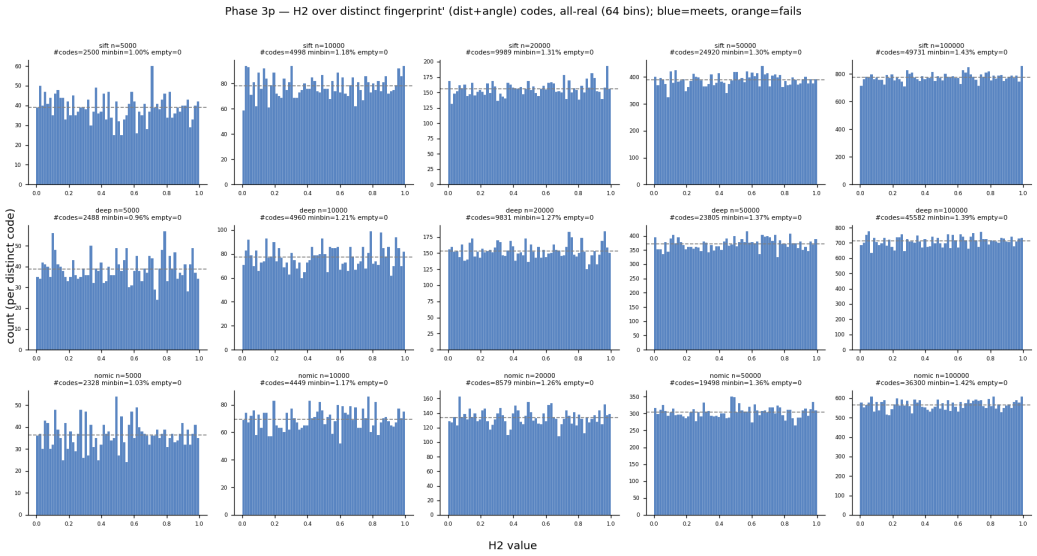


Figure 5: Distribution of $H_2$ values over distinct fingerprints in the candidate region. Rows show SIFT, DEEP, and nomic; columns show the database sizes indicated above the panels. Each fingerprint contributes once to one of 64 equal-width bins. Dashed lines show the expected count under uniform assignment.

Table 11: Complete attack results on SIFT over 50 trials at $\Delta_\theta = 0.12$. BER is mean $\pm$ standard deviation over all 64 positions, with erasures counted as errors. AUROC and TPR use 5,000 unwatermarked scores. TPR subscripts specify FPR. Cost is mean recall@10 loss in percentage points relative to the original database, including embedding.

| Attack | BER | AUROC | $\text{TPR}_{1\%}$ | $\text{TPR}_{0.5\%}$ | $\text{TPR}_{0.1\%}$ | Cost |
|---|---|---|---|---|---|---|
| No attack | 0.007±0.010 | **1.000** | **1.00** | **1.00** | **1.00** | 0.3 |
| Rotation | 0.007±0.010 | **1.000** | **1.00** | **1.00** | **1.00** | 0.3 |
| Reflection | 0.007±0.011 | **1.000** | **1.00** | **1.00** | **1.00** | 0.3 |
| Coordinate permutation | 0.007±0.010 | **1.000** | **1.00** | **1.00** | **1.00** | 0.3 |
| Sign changes | 0.008±0.011 | **1.000** | **1.00** | **1.00** | **1.00** | 0.3 |
| Translation | 0.007±0.010 | **1.000** | **1.00** | **1.00** | **1.00** | 0.3 |
| Uniform scaling | 0.007±0.010 | **1.000** | **1.00** | **1.00** | **1.00** | 0.3 |
| C1 | 0.007±0.010 | **1.000** | **1.00** | **1.00** | **1.00** | 0.3 |
| C2 | 0.122±0.032 | **1.000** | **1.00** | **1.00** | **1.00** | 1.4 |
| C3 | 0.189±0.046 | **1.000** | **1.00** | **1.00** | 0.98 | 10.1 |
| C4 | 0.046±0.029 | **1.000** | **1.00** | **1.00** | **1.00** | 17.2 |
| C5 | 0.340±0.051 | 0.994 | 0.92 | 0.80 | 0.52 | 10.5 |
| C6 | 0.007±0.011 | **1.000** | **1.00** | **1.00** | **1.00** | 0.3 |
| Shear, 2% | 0.177±0.052 | **1.000** | **1.00** | **1.00** | **1.00** | 2.4 |
| Outlier deletion, 10% | 0.020±0.017 | **1.000** | **1.00** | **1.00** | **1.00** | 8.0 |
| Outlier deletion, 20% | 0.045±0.031 | **1.000** | **1.00** | **1.00** | **1.00** | 17.2 |
| Outlier deletion, 40% | 0.150±0.039 | **1.000** | **1.00** | **1.00** | **1.00** | 36.6 |
| Random deletion, 10% | 0.190±0.047 | **1.000** | **1.00** | **1.00** | 0.98 | 10.1 |
| Random deletion, 20% | 0.404±0.068 | 0.939 | 0.50 | 0.36 | 0.20 | 20.0 |
| Gaussian noise, 0.064% | 0.037±0.025 | **1.000** | **1.00** | **1.00** | **1.00** | 0.7 |
| Gaussian noise, 0.161% | 0.122±0.034 | **1.000** | **1.00** | **1.00** | **1.00** | 1.4 |
| Uniform noise, 0.064% | 0.039±0.019 | **1.000** | **1.00** | **1.00** | **1.00** | 0.7 |
| Uniform noise, 0.161% | 0.129±0.038 | **1.000** | **1.00** | **1.00** | **1.00** | 1.3 |
| Random insertion, 30% | 0.022±0.017 | **1.000** | **1.00** | **1.00** | **1.00** | 0.3 |
| Adversarial insertion, 10% | 0.186±0.040 | **1.000** | **1.00** | **1.00** | **1.00** | 8.7 |
| Shuffle | 0.007±0.010 | **1.000** | **1.00** | **1.00** | **1.00** | 0.3 |
| Index rebuild | 0.008±0.011 | **1.000** | **1.00** | **1.00** | **1.00** | 0.3 |
| Float16 | 0.008±0.010 | **1.000** | **1.00** | **1.00** | **1.00** | 0.3 |
| 8-bit quantization | 0.142±0.043 | **1.000** | **1.00** | **1.00** | **1.00** | 1.5 |
| 6-bit quantization | 0.485±0.060 | 0.769 | 0.08 | 0.04 | 0.00 | 4.8 |
| Transparency deletion, 20% | 0.103±0.035 | **1.000** | **1.00** | **1.00** | **1.00** | 9.8 |
| Transparency deletion, 40% | 0.334±0.069 | 0.991 | 0.84 | 0.74 | 0.50 | 25.5 |
| Modify 30 coordinates, 20% | 0.131±0.042 | **1.000** | **1.00** | **1.00** | **1.00** | 9.4 |

Table 12: Complete attack results on DEEP over 50 trials at $\Delta_\theta = 0.12$. BER is mean $\pm$ standard deviation over all 64 positions, with erasures counted as errors. AUROC and TPR use 5,000 unwatermarked scores. TPR subscripts specify FPR. Cost is mean recall@10 loss in percentage points relative to the original database, including embedding. Costs marked † are affected by float32 errors under translation.

| Attack | BER | AUROC | $\text{TPR}_{1\%}$ | $\text{TPR}_{0.5\%}$ | $\text{TPR}_{0.1\%}$ | Cost |
|---|---|---|---|---|---|---|
| No attack | 0.016±0.015 | **1.000** | **1.00** | **1.00** | **1.00** | 0.4 |
| Rotation | 0.016±0.015 | **1.000** | **1.00** | **1.00** | **1.00** | 0.4 |
| Reflection | 0.016±0.015 | **1.000** | **1.00** | **1.00** | **1.00** | 0.4 |
| Coordinate permutation | 0.016±0.015 | **1.000** | **1.00** | **1.00** | **1.00** | 0.4 |
| Sign changes | 0.016±0.015 | **1.000** | **1.00** | **1.00** | **1.00** | 0.4 |
| Translation | 0.016±0.015 | **1.000** | **1.00** | **1.00** | **1.00** | 85.1† |
| Uniform scaling | 0.016±0.015 | **1.000** | **1.00** | **1.00** | **1.00** | 0.4 |
| C1 | 0.016±0.015 | **1.000** | **1.00** | **1.00** | **1.00** | 4.5† |
| C2 | 0.244±0.072 | 1.000 | **1.00** | 0.98 | 0.92 | 85.2† |
| C3 | 0.226±0.053 | **1.000** | **1.00** | **1.00** | **1.00** | 85.8† |
| C4 | 0.043±0.024 | **1.000** | **1.00** | **1.00** | **1.00** | 87.8† |
| C5 | 0.417±0.059 | 0.935 | 0.46 | 0.30 | 0.16 | 85.8† |
| C6 | 0.016±0.015 | **1.000** | **1.00** | **1.00** | **1.00** | 0.4 |
| Shear, 2% | 0.174±0.050 | **1.000** | **1.00** | **1.00** | **1.00** | 1.9 |
| Outlier deletion, 10% | 0.024±0.016 | **1.000** | **1.00** | **1.00** | **1.00** | 8.8 |
| Outlier deletion, 20% | 0.042±0.024 | **1.000** | **1.00** | **1.00** | **1.00** | 17.8 |
| Outlier deletion, 40% | 0.109±0.037 | **1.000** | **1.00** | **1.00** | **1.00** | 34.6 |
| Random deletion, 10% | 0.224±0.053 | **1.000** | **1.00** | **1.00** | **1.00** | 10.2 |
| Random deletion, 20% | 0.409±0.068 | 0.936 | 0.50 | 0.40 | 0.26 | 20.0 |
| Gaussian noise, 0.064% | 0.094±0.038 | **1.000** | **1.00** | **1.00** | **1.00** | 1.0 |
| Gaussian noise, 0.161% | 0.246±0.071 | 1.000 | **1.00** | 0.98 | 0.92 | 1.8 |
| Uniform noise, 0.064% | 0.081±0.028 | **1.000** | **1.00** | **1.00** | **1.00** | 1.0 |
| Uniform noise, 0.161% | 0.230±0.055 | 1.000 | **1.00** | 0.98 | 0.98 | 1.8 |
| Random insertion, 30% | 0.034±0.022 | **1.000** | **1.00** | **1.00** | **1.00** | 0.4 |
| Adversarial insertion, 10% | 0.201±0.047 | **1.000** | **1.00** | **1.00** | **1.00** | 8.6 |
| Shuffle | 0.015±0.015 | **1.000** | **1.00** | **1.00** | **1.00** | 0.4 |
| Index rebuild | 0.016±0.015 | **1.000** | **1.00** | **1.00** | **1.00** | 0.4 |
| Float16 | 0.016±0.014 | **1.000** | **1.00** | **1.00** | **1.00** | 0.4 |
| 8-bit quantization | 0.228±0.054 | **1.000** | **1.00** | **1.00** | **1.00** | 1.7 |
| 6-bit quantization | 0.533±0.059 | 0.555 | 0.02 | 0.00 | 0.00 | 5.6 |
| Transparency deletion, 20% | 0.082±0.030 | **1.000** | **1.00** | **1.00** | **1.00** | 9.2 |
| Transparency deletion, 40% | 0.266±0.053 | 1.000 | **1.00** | 0.98 | 0.94 | 22.4 |
| Modify 30 coordinates, 20% | 0.084±0.031 | **1.000** | **1.00** | **1.00** | **1.00** | 8.8 |

Table 13: Complete attack results on nomic over 50 trials at $\Delta_\theta = 0.12$. BER is mean $\pm$ standard deviation over all 64 positions, with erasures counted as errors. AUROC and TPR use 5,000 unwatermarked scores. TPR subscripts specify FPR. Cost is mean recall@10 loss in percentage points relative to the original database, including embedding. Costs marked † are affected by float32 errors under translation.

| Attack | BER | AUROC | $\text{TPR}_{1\%}$ | $\text{TPR}_{0.5\%}$ | $\text{TPR}_{0.1\%}$ | Cost |
|---|---|---|---|---|---|---|
| No attack | 0.021±0.017 | **1.000** | **1.00** | **1.00** | **1.00** | 0.6 |
| Rotation | 0.021±0.016 | **1.000** | **1.00** | **1.00** | **1.00** | 0.6 |
| Reflection | 0.021±0.016 | **1.000** | **1.00** | **1.00** | **1.00** | 0.6 |
| Coordinate permutation | 0.021±0.017 | **1.000** | **1.00** | **1.00** | **1.00** | 0.6 |
| Sign changes | 0.021±0.016 | **1.000** | **1.00** | **1.00** | **1.00** | 0.6 |
| Translation | 0.021±0.017 | **1.000** | **1.00** | **1.00** | **1.00** | 100.0† |
| Uniform scaling | 0.021±0.017 | **1.000** | **1.00** | **1.00** | **1.00** | 0.6 |
| C1 | 0.021±0.016 | **1.000** | **1.00** | **1.00** | **1.00** | 56.0† |
| C2 | 0.246±0.060 | 1.000 | **1.00** | **1.00** | 0.98 | 100.0† |
| C3 | 0.219±0.043 | **1.000** | **1.00** | **1.00** | **1.00** | 100.0† |
| C4 | 0.056±0.023 | **1.000** | **1.00** | **1.00** | **1.00** | 100.0† |
| C5 | 0.415±0.055 | 0.941 | 0.24 | 0.24 | 0.20 | 100.0† |
| C6 | 0.021±0.016 | **1.000** | **1.00** | **1.00** | **1.00** | 0.6 |
| Shear, 2% | 0.068±0.029 | **1.000** | **1.00** | **1.00** | **1.00** | 1.3 |
| Outlier deletion, 10% | 0.032±0.018 | **1.000** | **1.00** | **1.00** | **1.00** | 7.7 |
| Outlier deletion, 20% | 0.053±0.023 | **1.000** | **1.00** | **1.00** | **1.00** | 15.7 |
| Outlier deletion, 40% | 0.114±0.036 | **1.000** | **1.00** | **1.00** | **1.00** | 33.0 |
| Random deletion, 10% | 0.219±0.042 | **1.000** | **1.00** | **1.00** | **1.00** | 10.2 |
| Random deletion, 20% | 0.407±0.049 | 0.958 | 0.30 | 0.30 | 0.14 | 20.1 |
| Gaussian noise, 0.064% | 0.099±0.032 | **1.000** | **1.00** | **1.00** | **1.00** | 1.3 |
| Gaussian noise, 0.161% | 0.247±0.059 | 1.000 | 0.98 | 0.98 | 0.98 | 2.3 |
| Uniform noise, 0.064% | 0.091±0.031 | **1.000** | **1.00** | **1.00** | **1.00** | 1.3 |
| Uniform noise, 0.161% | 0.245±0.050 | **1.000** | **1.00** | **1.00** | 0.98 | 2.4 |
| Random insertion, 30% | 0.040±0.021 | **1.000** | **1.00** | **1.00** | **1.00** | 0.6 |
| Adversarial insertion, 10% | 0.196±0.051 | **1.000** | **1.00** | **1.00** | **1.00** | 8.3 |
| Shuffle | 0.021±0.016 | **1.000** | **1.00** | **1.00** | **1.00** | 0.6 |
| Index rebuild | 0.021±0.016 | **1.000** | **1.00** | **1.00** | **1.00** | 0.6 |
| Float16 | 0.021±0.017 | **1.000** | **1.00** | **1.00** | **1.00** | 0.6 |
| 8-bit quantization | 0.135±0.038 | **1.000** | **1.00** | **1.00** | **1.00** | 1.6 |
| 6-bit quantization | 0.465±0.068 | 0.805 | 0.12 | 0.12 | 0.04 | 4.5 |
| Transparency deletion, 20% | 0.088±0.031 | **1.000** | **1.00** | **1.00** | **1.00** | 8.8 |
| Transparency deletion, 40% | 0.235±0.049 | **1.000** | **1.00** | **1.00** | **1.00** | 21.6 |
| Modify 30 coordinates, 20% | 0.090±0.034 | **1.000** | **1.00** | **1.00** | **1.00** | 7.9 |

Table 14: Complete attack results on GIST over 50 trials at $\Delta_\theta = 0.12$. BER is mean $\pm$ standard deviation over all 64 positions, with erasures counted as errors. AUROC and TPR use 5,000 unwatermarked scores. TPR subscripts specify FPR. Cost is mean recall@10 loss in percentage points relative to the original database, including embedding. Costs marked † are affected by float32 errors under translation.

| Attack | BER | AUROC | $\text{TPR}_{1\%}$ | $\text{TPR}_{0.5\%}$ | $\text{TPR}_{0.1\%}$ | Cost |
|---|---|---|---|---|---|---|
| No attack | 0.130±0.036 | **1.000** | **1.00** | **1.00** | **1.00** | 1.8 |
| Rotation | 0.133±0.037 | **1.000** | **1.00** | **1.00** | **1.00** | 1.8 |
| Reflection | 0.132±0.035 | **1.000** | **1.00** | **1.00** | **1.00** | 1.8 |
| Coordinate permutation | 0.132±0.038 | **1.000** | **1.00** | **1.00** | **1.00** | 1.8 |
| Sign changes | 0.131±0.037 | **1.000** | **1.00** | **1.00** | **1.00** | 1.8 |
| Translation | 0.131±0.038 | **1.000** | **1.00** | **1.00** | **1.00** | 99.9† |
| Uniform scaling | 0.133±0.035 | **1.000** | **1.00** | **1.00** | **1.00** | 1.8 |
| C1 | 0.136±0.035 | **1.000** | **1.00** | **1.00** | **1.00** | 41.7† |
| C2 | 0.440±0.059 | 0.899 | 0.32 | 0.18 | 0.10 | 99.9† |
| C3 | 0.342±0.055 | 0.994 | 0.84 | 0.74 | 0.68 | 99.9† |
| C4 | 0.189±0.043 | **1.000** | **1.00** | **1.00** | **1.00** | 100.0† |
| C5 | 0.486±0.066 | 0.768 | 0.08 | 0.04 | 0.04 | 99.9† |
| C6 | 0.142±0.044 | **1.000** | **1.00** | **1.00** | **1.00** | 1.9 |
| Shear, 2% | 0.214±0.048 | **1.000** | **1.00** | **1.00** | **1.00** | 2.7 |
| Outlier deletion, 10% | 0.170±0.041 | **1.000** | **1.00** | **1.00** | **1.00** | 4.5 |
| Outlier deletion, 20% | 0.188±0.042 | **1.000** | **1.00** | **1.00** | **1.00** | 8.6 |
| Outlier deletion, 40% | 0.265±0.056 | 1.000 | **1.00** | **1.00** | 0.96 | 19.4 |
| Random deletion, 10% | 0.339±0.052 | 0.997 | 0.94 | 0.74 | 0.64 | 10.9 |
| Random deletion, 20% | 0.460±0.075 | 0.812 | 0.32 | 0.12 | 0.08 | 20.4 |
| Gaussian noise, 0.064% | 0.271±0.067 | 0.999 | 0.96 | 0.96 | 0.94 | 2.9 |
| Gaussian noise, 0.161% | 0.440±0.053 | 0.908 | 0.26 | 0.20 | 0.10 | 4.4 |
| Uniform noise, 0.064% | 0.287±0.049 | 0.999 | **1.00** | 0.94 | 0.90 | 2.9 |
| Uniform noise, 0.161% | 0.431±0.067 | 0.905 | 0.36 | 0.22 | 0.16 | 4.4 |
| Random insertion, 30% | 0.293±0.055 | 0.999 | 0.98 | 0.94 | 0.88 | 1.8 |
| Adversarial insertion, 10% | 0.328±0.051 | 0.997 | 0.92 | 0.84 | 0.70 | 8.9 |
| Shuffle | 0.137±0.041 | **1.000** | **1.00** | **1.00** | **1.00** | 1.9 |
| Index rebuild | 0.131±0.036 | **1.000** | **1.00** | **1.00** | **1.00** | 1.8 |
| Float16 | 0.140±0.037 | **1.000** | **1.00** | **1.00** | **1.00** | 1.9 |
| 8-bit quantization | 0.386±0.050 | 0.979 | 0.68 | 0.48 | 0.30 | 3.8 |
| 6-bit quantization | 0.544±0.063 | 0.520 | 0.02 | 0.00 | 0.00 | 10.5 |
| Transparency deletion, 20% | 0.191±0.044 | **1.000** | **1.00** | **1.00** | **1.00** | 4.2 |
| Transparency deletion, 40% | 0.253±0.055 | 1.000 | **1.00** | **1.00** | 0.98 | 9.5 |
| Modify 30 coordinates, 20% | 0.151±0.039 | **1.000** | **1.00** | **1.00** | **1.00** | 3.9 |

Table 15: Matched exact-search and HNSW measurements at $n = 60{,}000$ with $\Delta_\theta = 0.12$, $k = 8$, and one common key and data split. Embedding and detection times are in seconds and include index construction. Each is the median of three timing repetitions on the same node for the two backends within a dataset, using 16 CPU threads. HNSW uses $M = 32$, $\text{efConstruction} = 200$, and $\text{efSearch} = 128$. The speed-up columns compare these matched runs. BER includes erasures; cost is recall@10 loss in percentage points, measured by exact retrieval. These timing repetitions are distinct from the 50-trial robustness evaluation.

| Dataset | Search | Embed | Detect | Emb.× | Det.× | BER | Cost |
|---|---|---|---|---|---|---|---|
| SIFT | Exact | 30.44 | 33.30 | – | – | 0.000 | 0.220 |
| | HNSW | 1.32 | 1.73 | 23.1 | 19.2 | 0.000 | 0.220 |
| DEEP | Exact | 29.77 | 33.59 | – | – | 0.000 | 0.335 |
| | HNSW | 1.21 | 1.44 | 24.7 | 23.3 | 0.000 | 0.335 |
| nomic | Exact | 65.22 | 62.73 | – | – | 0.000 | 0.355 |
| | HNSW | 45.70 | 44.80 | 1.4 | 1.4 | 0.000 | 0.355 |
| GIST | Exact | 46.98 | 50.77 | – | – | 0.094 | 1.800 |
| | HNSW | 23.22 | 22.79 | 2.0 | 2.2 | 0.125 | 1.845 |

Table 16: Wall time in seconds across database sizes. Each cell lists exact search / HNSW. All runs use the final configuration, target anchor budget $A = 1122$, one fixed key, and 16 CPU threads. Databases are nested prefixes of genuine data; query and calibration slices are disjoint from every database prefix. Medians of three timing repetitions are reported, except exact-search values marked †, which use one repetition. The scaling jobs ran on different cluster nodes, so the ratios are not controlled estimates of backend speed-up. The matched comparison is in Table 15.

| Dataset | Size | Embed (exact / HNSW) | Detect (exact / HNSW) |
|---|---|---|---|
| SIFT | 15k | 2.6 / 0.4 | 2.9 / 0.4 |
| | 30k | 15.6 / 0.7 | 14.5 / 0.8 |
| | 60k | 40.5 / 1.6 | 40.7 / 2.0 |
| | 120k | 129.5 / 6.7 | 127.9 / 8.9 |
| | 250k | 503.3† / 17.1 | 473.8† / 21.0 |
| DEEP | 15k | 3.1 / 0.4 | 3.0 / 0.5 |
| | 30k | 8.9 / 0.6 | 10.7 / 0.7 |
| | 60k | 35.3 / 1.8 | 39.6 / 1.9 |
| | 120k | 115.1 / 4.2 | 112.5 / 4.5 |
| | 250k | 489.8† / 13.0 | 522.4† / 15.9 |
| nomic | 15k | 3.3 / 2.1 | 4.4 / 3.1 |
| | 30k | 12.9 / 5.8 | 15.1 / 8.1 |
| | 60k | 72.9 / 19.1 | 72.4 / 27.1 |
| | 120k | 174.0† / 63.3 | 179.2† / 55.3 |
| | 250k | 847.0† / 98.1 | 950.1† / 126.4 |
| GIST | 15k | 5.4 / 2.3 | 6.1 / 3.7 |
| | 30k | 16.8 / 6.2 | 19.0 / 8.8 |
| | 60k | 52.8 / 17.0 | 54.5 / 22.9 |
| | 120k | 157.2† / 55.8 | 168.6† / 64.9 |
| | 250k | 727.2† / 104.7 | 808.7† / 131.2 |

Table 17: Recovery and retrieval quality for the scaling measurements in Table 16. Each cell lists exact search / HNSW. BER counts errors and erasures over all 64 message positions. Cost is recall@10 loss in percentage points. $A_{\text{eff}}$ counts distinct fingerprints among the selected anchors. Results use one fixed key per setting; timing repetitions do not provide independent recovery trials.

| Dataset | Size | BER | Cost | $A_{\text{eff}}$ |
|---|---|---|---|---|
| SIFT | 15k | 0.047 / 0.047 | 0.570 / 0.570 | 1120 / 1120 |
| | 30k | 0.000 / 0.000 | 0.360 / 0.360 | 1119 / 1119 |
| | 60k | 0.000 / 0.000 | 0.220 / 0.220 | 1120 / 1120 |
| | 120k | 0.000 / 0.000 | 0.085 / 0.085 | 1114 / 1114 |
| | 250k | 0.000 / 0.000 | 0.075 / 0.075 | 1102 / 1102 |
| DEEP | 15k | 0.063 / 0.063 | 0.680 / 0.680 | 1111 / 1111 |
| | 30k | 0.031 / 0.031 | 0.510 / 0.510 | 1099 / 1099 |
| | 60k | 0.000 / 0.000 | 0.335 / 0.335 | 1055 / 1055 |
| | 120k | 0.000 / 0.000 | 0.125 / 0.125 | 1020 / 1019 |
| | 250k | 0.016 / 0.016 | 0.090 / 0.090 | 898 / 895 |
| nomic | 15k | 0.094 / 0.094 | 1.240 / 1.240 | 953 / 952 |
| | 30k | 0.063 / 0.063 | 0.830 / 0.830 | 921 / 921 |
| | 60k | 0.000 / 0.000 | 0.355 / 0.355 | 869 / 869 |
| | 120k | 0.016 / 0.016 | 0.225 / 0.220 | 799 / 797 |
| | 250k | 0.016 / 0.016 | 0.095 / 0.095 | 726 / 727 |
| GIST | 15k | 0.219 / 0.219 | 2.300 / 2.285 | 962 / 967 |
| | 30k | 0.078 / 0.141 | 2.125 / 2.125 | 777 / 779 |
| | 60k | 0.094 / 0.156 | 1.800 / 1.755 | 693 / 699 |
| | 120k | 0.078 / 0.094 | 1.195 / 1.255 | 470 / 470 |
| | 250k | 0.063 / 0.109 | 0.750 / 0.765 | 399 / 411 |

Table 18: Algorithm-aware angle attacks on SIFT and DEEP at $\Delta_\theta = 0.12$. Whole attacks every center; Region restricts centers to the inner selection region. The two scopes are paired over 10 trials of 42,000 vectors. TPR is measured at 1% FPR using 1,000 unwatermarked scores per dataset, shared across strengths and scopes. BER and cost are mean ± standard deviation; erasures count as bit errors. Cost is recall@10 loss in percentage points relative to the original database, including embedding. The no-attack row records embedding cost alone.

| Dataset | Scope | $\rho$ | BER | AUROC | TPR | Cost |
|---|---|---|---|---|---|---|
| SIFT | None | 0.00 | 0.011±0.011 | 1.000 | 1.00 | 0.30±0.04 |
| | Whole | 0.25 | 0.491±0.080 | 0.713 | 0.10 | 4.59±0.10 |
| | Whole | 0.50 | 0.538±0.052 | 0.543 | 0.00 | 8.60±0.11 |
| | Whole | 0.75 | 0.555±0.056 | 0.457 | 0.00 | 12.80±0.19 |
| | Whole | 1.00 | 0.591±0.070 | 0.299 | 0.00 | 17.07±0.18 |
| | Region | 0.25 | 0.459±0.065 | 0.844 | 0.10 | 3.39±0.13 |
| | Region | 0.50 | 0.506±0.058 | 0.678 | 0.00 | 6.29±0.15 |
| | Region | 0.75 | 0.561±0.041 | 0.419 | 0.00 | 9.26±0.08 |
| | Region | 1.00 | 0.531±0.079 | 0.533 | 0.10 | 12.32±0.12 |
| DEEP | None | 0.00 | 0.013±0.014 | 1.000 | 1.00 | 0.38±0.04 |
| | Whole | 0.25 | 0.500±0.059 | 0.696 | 0.00 | 5.42±0.13 |
| | Whole | 0.50 | 0.566±0.073 | 0.422 | 0.00 | 10.09±0.16 |
| | Whole | 0.75 | 0.552±0.059 | 0.486 | 0.00 | 14.87±0.18 |
| | Whole | 1.00 | 0.592±0.057 | 0.289 | 0.00 | 19.96±0.23 |
| | Region | 0.25 | 0.506±0.059 | 0.683 | 0.00 | 3.81±0.11 |
| | Region | 0.50 | 0.559±0.052 | 0.446 | 0.00 | 7.04±0.10 |
| | Region | 0.75 | 0.567±0.077 | 0.413 | 0.00 | 10.28±0.13 |
| | Region | 1.00 | 0.552±0.061 | 0.502 | 0.00 | 13.71±0.18 |

Table 19: Algorithm-aware angle attacks on nomic and GIST, using the same paired 10-trial protocol as Table 18. TPR is measured at 1% FPR using 1,000 unwatermarked scores per dataset. BER and recall@10 loss in percentage points are mean ± standard deviation. Region restricts attacked centers, while selected neighbours may lie outside that region.

| Dataset | Scope | $\rho$ | BER | AUROC | TPR | Cost |
|---|---|---|---|---|---|---|
| nomic | None | 0.00 | 0.020±0.015 | 1.000 | 1.00 | 0.59±0.07 |
| | Whole | 0.25 | 0.333±0.044 | 0.997 | 0.90 | 3.03±0.10 |
| | Whole | 0.50 | 0.488±0.062 | 0.758 | 0.20 | 6.17±0.13 |
| | Whole | 0.75 | 0.530±0.053 | 0.584 | 0.00 | 10.39±0.12 |
| | Whole | 1.00 | 0.534±0.080 | 0.509 | 0.10 | 15.47±0.16 |
| | Region | 0.25 | 0.266±0.045 | 1.000 | 1.00 | 2.18±0.12 |
| | Region | 0.50 | 0.444±0.082 | 0.854 | 0.30 | 4.20±0.14 |
| | Region | 0.75 | 0.527±0.053 | 0.633 | 0.00 | 6.81±0.19 |
| | Region | 1.00 | 0.575±0.050 | 0.349 | 0.00 | 10.01±0.19 |
| GIST | None | 0.00 | 0.128±0.043 | 1.000 | 1.00 | 1.79±0.11 |
| | Whole | 0.25 | 0.528±0.059 | 0.602 | 0.00 | 8.72±0.20 |
| | Whole | 0.50 | 0.538±0.063 | 0.558 | 0.00 | 20.23±0.26 |
| | Whole | 0.75 | 0.566±0.083 | 0.479 | 0.00 | 32.32±0.24 |
| | Whole | 1.00 | 0.539±0.058 | 0.572 | 0.00 | 42.97±0.30 |
| | Region | 0.25 | 0.498±0.038 | 0.776 | 0.00 | 6.58±0.20 |
| | Region | 0.50 | 0.563±0.077 | 0.443 | 0.10 | 14.78±0.14 |
| | Region | 0.75 | 0.533±0.062 | 0.597 | 0.00 | 23.25±0.29 |
| | Region | 1.00 | 0.538±0.086 | 0.534 | 0.00 | 30.65±0.30 |

Table 20: Effect of neighbourhood size on the final method. Entries are mean message-bit accuracy over 50 trials per dataset and setting; higher is better. Empty or tied message positions count as errors. The remaining parameters are fixed at $q_d = 0.08$, $q_\theta = 0.05$, and $\Delta_\theta = 0.12$. Del. denotes random deletion. Noise is Gaussian with relative scale 0.161%. The final column uses 8-bit coordinate quantization. The default is $k = 8$.

| Dataset | $k$ | Clean | Rotation | Del. 10% | Del. 20% | Noise | 8-bit |
|---|---|---|---|---|---|---|---|
| SIFT | 4 | 0.968 | 0.968 | 0.870 | 0.736 | 0.909 | 0.911 |
| | 6 | 0.988 | 0.988 | 0.848 | 0.650 | 0.912 | 0.917 |
| | 8 | 0.993 | 0.993 | 0.809 | 0.596 | 0.876 | 0.859 |
| | 12 | 0.988 | 0.988 | 0.694 | 0.497 | 0.761 | 0.752 |
| | 16 | 0.990 | 0.990 | 0.614 | 0.477 | 0.692 | 0.676 |
| DEEP | 4 | 0.869 | 0.869 | 0.765 | 0.660 | 0.796 | 0.787 |
| | 6 | 0.976 | 0.975 | 0.831 | 0.633 | 0.833 | 0.828 |
| | 8 | 0.984 | 0.984 | 0.777 | 0.590 | 0.754 | 0.771 |
| | 12 | 0.974 | 0.974 | 0.664 | 0.494 | 0.644 | 0.650 |
| | 16 | 0.975 | 0.975 | 0.585 | 0.474 | 0.565 | 0.584 |
| nomic | 4 | 0.797 | 0.797 | 0.695 | 0.598 | 0.718 | 0.751 |
| | 6 | 0.959 | 0.959 | 0.819 | 0.649 | 0.802 | 0.880 |
| | 8 | 0.979 | 0.979 | 0.782 | 0.593 | 0.753 | 0.865 |
| | 12 | 0.979 | 0.979 | 0.676 | 0.495 | 0.632 | 0.767 |
| | 16 | 0.969 | 0.969 | 0.588 | 0.458 | 0.568 | 0.708 |
| GIST | 4 | 0.702 | 0.708 | 0.618 | 0.548 | 0.577 | 0.618 |
| | 6 | 0.841 | 0.843 | 0.685 | 0.573 | 0.619 | 0.650 |
| | 8 | 0.866 | 0.864 | 0.668 | 0.547 | 0.566 | 0.619 |
| | 12 | 0.827 | 0.827 | 0.583 | 0.482 | 0.508 | 0.537 |
| | 16 | 0.782 | 0.780 | 0.510 | 0.429 | 0.485 | 0.502 |

Table 21: Sensitivity to fingerprint resolution and encoding strength. Entries are mean message-bit accuracy over 50 trials; higher is better. Empty or tied positions count as errors. The default configuration uses $k = 8$, $q_d = 0.08$, $q_\theta = 0.05$, and $\Delta_\theta = 0.12$. Each other row changes only the indicated parameter. Angular parameters are in radians. Del. denotes random deletion. Noise is Gaussian with relative scale 0.161%. The final column uses 8-bit coordinate quantization.

| Dataset | Setting | Clean | Rotation | Del. 10% | Del. 20% | Noise | 8-bit |
|---|---|---|---|---|---|---|---|
| SIFT | Default | 0.993 | 0.993 | 0.809 | 0.596 | 0.876 | 0.859 |
| | $q_d = 0.04$ | 0.992 | 0.991 | 0.799 | 0.583 | 0.826 | 0.808 |
| | $q_d = 0.12$ | 0.990 | 0.990 | 0.794 | 0.589 | 0.880 | 0.868 |
| | $q_\theta = 0.025$ | 0.962 | 0.962 | 0.731 | 0.552 | 0.728 | 0.718 |
| | $q_\theta = 0.075$ | 0.997 | 0.997 | 0.859 | 0.629 | 0.922 | 0.916 |
| | $\Delta_\theta = 0.06$ | 0.999 | 0.999 | 0.853 | 0.627 | 0.919 | 0.906 |
| | $\Delta_\theta = 0.09$ | 0.995 | 0.995 | 0.826 | 0.591 | 0.908 | 0.886 |
| | $\Delta_\theta = 0.15$ | 0.982 | 0.982 | 0.777 | 0.570 | 0.844 | 0.832 |
| DEEP | Default | 0.984 | 0.984 | 0.777 | 0.590 | 0.754 | 0.771 |
| | $q_d = 0.04$ | 0.983 | 0.983 | 0.768 | 0.582 | 0.708 | 0.731 |
| | $q_d = 0.12$ | 0.982 | 0.982 | 0.777 | 0.577 | 0.777 | 0.777 |
| | $q_\theta = 0.025$ | 0.929 | 0.929 | 0.708 | 0.543 | 0.627 | 0.613 |
| | $q_\theta = 0.075$ | 0.989 | 0.989 | 0.808 | 0.586 | 0.826 | 0.837 |
| | $\Delta_\theta = 0.06$ | 0.996 | 0.996 | 0.842 | 0.613 | 0.815 | 0.837 |
| | $\Delta_\theta = 0.09$ | 0.992 | 0.992 | 0.809 | 0.599 | 0.782 | 0.804 |
| | $\Delta_\theta = 0.15$ | 0.969 | 0.969 | 0.748 | 0.567 | 0.740 | 0.749 |
| nomic | Default | 0.979 | 0.979 | 0.782 | 0.593 | 0.753 | 0.865 |
| | $q_d = 0.04$ | 0.981 | 0.981 | 0.787 | 0.588 | 0.712 | 0.844 |
| | $q_d = 0.12$ | 0.977 | 0.977 | 0.791 | 0.588 | 0.777 | 0.869 |
| | $q_\theta = 0.025$ | 0.935 | 0.935 | 0.703 | 0.549 | 0.619 | 0.729 |
| | $q_\theta = 0.075$ | 0.979 | 0.979 | 0.800 | 0.606 | 0.794 | 0.904 |
| | $\Delta_\theta = 0.06$ | 0.995 | 0.995 | 0.843 | 0.618 | 0.805 | 0.927 |
| | $\Delta_\theta = 0.09$ | 0.986 | 0.986 | 0.816 | 0.607 | 0.780 | 0.893 |
| | $\Delta_\theta = 0.15$ | 0.965 | 0.965 | 0.762 | 0.590 | 0.722 | 0.833 |
| GIST | Default | 0.866 | 0.864 | 0.668 | 0.547 | 0.566 | 0.619 |
| | $q_d = 0.04$ | 0.862 | 0.859 | 0.665 | 0.552 | 0.568 | 0.584 |
| | $q_d = 0.12$ | 0.861 | 0.862 | 0.671 | 0.539 | 0.587 | 0.628 |
| | $q_\theta = 0.025$ | 0.780 | 0.781 | 0.599 | 0.502 | 0.509 | 0.553 |
| | $q_\theta = 0.075$ | 0.871 | 0.874 | 0.660 | 0.534 | 0.593 | 0.645 |
| | $\Delta_\theta = 0.06$ | 0.945 | 0.946 | 0.727 | 0.551 | 0.618 | 0.671 |
| | $\Delta_\theta = 0.09$ | 0.913 | 0.914 | 0.685 | 0.556 | 0.602 | 0.660 |
| | $\Delta_\theta = 0.15$ | 0.826 | 0.828 | 0.639 | 0.525 | 0.553 | 0.596 |

Table 22: Spread of carrier angles and nearest-neighbour distance ratios in a diagnostic run with 60,000 vectors per dataset. Normalized entropy is the Shannon entropy of a 64-bin histogram over the observed support, divided by $\log 64$. Angle IQR is measured in radians; distance-ratio IQR is dimensionless.

| | Carrier angle | | $d_2/d_1$ | |
|---|---|---|---|---|
| Dataset | Entropy | IQR | Entropy | IQR |
| SIFT | 0.886 | 0.431 | 0.328 | 0.452 |
| DEEP | 0.834 | 0.238 | 0.543 | 0.051 |
| nomic | 0.771 | 0.164 | 0.042 | 0.049 |
| GIST | 0.761 | 0.176 | 0.003 | 0.030 |

# E WHY THE MESSAGE IS ENCODED IN ANGLES

Both distance ratios and angles are invariant to similarity transforms. This appendix reports neighbour-distance concentration, feature spread, and fingerprint distinctness. These measurements describe the local geometry used by the watermark.

**Neighbour-distance concentration.** The median ratio between the second and first nearest-neighbour distances is $1.172$ on SIFT, compared with $1.016$–$1.029$ on DEEP, nomic, and GIST (Table 23). Ratios close to one indicate that the nearest neighbours lie at similar distances.

Encoding through a distance change can therefore alter neighbour order, which also affects the fingerprint used at detection. Angle encoding allows a different movement. A neighbour is rotated around the anchor while its distance to that anchor remains fixed.

**Displacement and readability.** To write a bit, the encoded quantity is moved to the nearest quantization grid point carrying the desired label. The required displacement depends on the initial value, the grid spacing, and the keyed offset.

A smaller encoding step reduces the required change but also reduces the margin for reading the bit after further perturbations. A larger encoding step increases that margin while requiring larger embedding changes. The relevant question is whether the chosen quantity supports reliable recovery while preserving retrieval quality.

**Feature spread.** Table 22 reports normalized histogram entropy and interquartile range for the carrier angle and the ratio $d_2/d_1$. The angle distributions have normalized entropy between approximately $0.76$ and $0.89$.

The distance-ratio IQR is much smaller on DEEP, nomic, and GIST than on SIFT. Entropy and IQR describe different aspects of these distributions. Entropy measures occupancy across histogram bins, while IQR measures the width of the central half of the observations. Because the histograms use their observed supports, a narrow distribution can still have high normalized entropy, as on DEEP.

**Fingerprint distinctness and stability.** Fine fingerprint quantization distinguishes more vectors, but small embedding changes can cross quantization boundaries. Coarser quantization absorbs larger changes while merging more vectors into identical fingerprints.

Identical fingerprints receive identical keyed assignments. The effective number of independent message-position assignments is therefore the number of distinct fingerprints. Theorem 1 gives 561 as a sufficient count for its stated coverage bound.

Table 23 shows how adding angular features increases the number of distinct fingerprints at the tested quantization settings. On nomic, adding angles increases the number of distinct fingerprints from 38 to 869. In that diagnostic run, the combined fingerprints exceed the sufficient coverage count on all four datasets in this measurement.

Table 23: Neighbour-distance concentration and fingerprint distinctness, measured on 60,000 vectors per dataset. The final two columns count distinct fingerprints among the selected anchors, using distance ratios alone or distance ratios together with angles. Both constructions use $q_d = 0.08$; the combined fingerprint also uses $q_\theta = 0.05$. Counts are measured before embedding in one run per dataset.

| Dataset | $d$ | Median $d_2/d_1$ | Distance only | Distance + angle |
|---|---|---|---|---|
| GIST | 960 | 1.016 | 254 | 693 |
| nomic | 768 | 1.028 | 38 | 869 |
| DEEP | 96 | 1.029 | 70 | 1055 |
| SIFT | 128 | 1.172 | 244 | 1120 |

Table 24: Baseline detection thresholds and achieved empirical FPR. Each row uses 5,000 clean negative scores. FPR is a fraction.

| Dataset | Method | Threshold | Achieved FPR |
|---|---|---|---|
| SIFT | TVP | 0.62500 | 0.0036 |
| SIFT | RS | 0.62500 | 0.0036 |
| SIFT | SCPW | 0.65625 | 0.0000 |
| SIFT | TabularMark | 0.81000 | 0.0100 |
| DEEP | TVP | 0.67188 | 0.0084 |
| DEEP | RS | 0.67188 | 0.0084 |
| DEEP | SCPW | 0.68750 | 0.0054 |
| DEEP | TabularMark | 0.81000 | 0.0100 |
| nomic | TVP | 0.64062 | 0.0056 |
| nomic | RS | 0.64062 | 0.0056 |
| nomic | SCPW | 0.71875 | 0.0052 |
| nomic | TabularMark | 0.81000 | 0.0100 |
| GIST | TVP | 0.65625 | 0.0060 |
| GIST | RS | 0.65625 | 0.0060 |
| GIST | SCPW | 0.75000 | 0.0076 |
| GIST | TabularMark | 0.82000 | 0.0100 |

# F Complete baseline attack results

The following tables report all 33 conditions: one clean condition and 32 attacks or database transformations. Each functional baseline–dataset pair has 50 trials per condition and 5,000 clean negative scores. A trial uses a fresh seed and a 70% subsample of the 60,000-vector pool. The same clean negatives are reused across attack conditions. Calibration and reported FPR use these same negatives; this evaluation has no separate calibration split. The fixed rescaling and payload lengths are given in Section 5.1. Attack names follow Appendix D.

TVP, RS, and SCPW are reported using BER, with standard deviation across trials. TabularMark uses MP with its standard deviation. AUROC and TPR use bit accuracy for the message-based methods and MP for TabularMark. Thresholds are calibrated separately for each method and dataset at a nominal 1% FPR. Their achieved empirical FPRs are reported below; discrete scores need not attain exactly 1%.

Table 25: SIFT: complete attack results for TVP and RS. Recovery is mean±standard deviation over 50 trials. TPR uses a nominal 1% FPR. Thresholds and achieved FPR appear in Table 24.

| | TVP | | | RS | | |
|---|---|---|---|---|---|---|
| Condition | BER | AUROC | TPR | BER | AUROC | TPR |
| No attack | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Rotation | 0.512±0.059 | 0.459 | 0.00 | 0.518±0.051 | 0.425 | 0.00 |
| Reflection | 0.501±0.066 | 0.503 | 0.02 | 0.494±0.059 | 0.531 | 0.04 |
| Axis permutation | 0.508±0.059 | 0.480 | 0.00 | 0.507±0.058 | 0.485 | 0.00 |
| Sign flips | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Translation | 0.508±0.052 | 0.474 | 0.00 | 0.507±0.052 | 0.478 | 0.00 |
| Uniform scaling | 0.507±0.062 | 0.500 | 0.00 | 0.506±0.063 | 0.507 | 0.00 |
| C1 | 0.502±0.067 | 0.505 | 0.02 | 0.500±0.066 | 0.523 | 0.02 |
| C2 | 0.500±0.069 | 0.524 | 0.02 | 0.493±0.059 | 0.529 | 0.02 |
| C3 | 0.518±0.065 | 0.425 | 0.02 | 0.505±0.069 | 0.489 | 0.02 |
| C4 | 0.512±0.062 | 0.469 | 0.00 | 0.509±0.061 | 0.480 | 0.00 |
| C5 | 0.494±0.060 | 0.536 | 0.02 | 0.494±0.065 | 0.532 | 0.06 |
| C6 | 0.512±0.059 | 0.459 | 0.00 | 0.518±0.051 | 0.425 | 0.00 |
| Shear 2% | 0.473±0.064 | 0.646 | 0.04 | 0.469±0.069 | 0.654 | 0.06 |
| Extremal deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Extremal deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Extremal deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Random deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Random deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Gaussian 0.064% | 0.513±0.051 | 0.450 | 0.00 | 0.510±0.056 | 0.459 | 0.00 |
| Gaussian 0.161% | 0.497±0.056 | 0.529 | 0.00 | 0.509±0.063 | 0.466 | 0.02 |
| Uniform noise 0.064% | 0.501±0.067 | 0.517 | 0.02 | 0.498±0.068 | 0.542 | 0.00 |
| Uniform noise 0.161% | 0.503±0.049 | 0.503 | 0.00 | 0.504±0.056 | 0.491 | 0.00 |
| Random insertion 30% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Adversarial insertion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Shuffle | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Index rebuild | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Float16 | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| 8-bit quantization | 0.493±0.060 | 0.555 | 0.00 | 0.489±0.059 | 0.579 | 0.00 |
| 6-bit quantization | 0.512±0.064 | 0.453 | 0.02 | 0.510±0.063 | 0.459 | 0.02 |
| Transparency deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Transparency deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Modify30 (0.2) | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |

Table 26: SIFT: complete attack results for SCPW and TabularMark. Recovery is mean±standard deviation over 50 trials. TPR uses a nominal 1% FPR. TabularMark reports MP. Thresholds and achieved FPR appear in Table 24.

| | SCPW | | | TabularMark | | |
|---|---|---|---|---|---|---|
| Condition | BER | AUROC | TPR | MP | AUROC | TPR |
| No attack | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Rotation | 0.508±0.074 | 0.559 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Reflection | 0.489±0.097 | 0.603 | 0.04 | 0.000±0.000 | 0.000 | 0.00 |
| Axis permutation | 0.522±0.083 | 0.500 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Sign flips | 0.488±0.082 | 0.629 | 0.02 | 1.000±0.000 | 1.000 | 1.00 |
| Translation | 0.522±0.083 | 0.500 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Uniform scaling | 0.522±0.083 | 0.500 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C1 | 0.508±0.097 | 0.521 | 0.04 | 0.000±0.000 | 0.000 | 0.00 |
| C2 | 0.517±0.082 | 0.525 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C3 | 0.481±0.084 | 0.644 | 0.04 | 0.000±0.000 | 0.000 | 0.00 |
| C4 | 0.486±0.074 | 0.640 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C5 | 0.518±0.092 | 0.522 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C6 | 0.508±0.074 | 0.559 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Shear 2% | 0.522±0.083 | 0.500 | 0.00 | 0.020±0.000 | 0.000 | 0.00 |
| Extremal deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.890±0.000 | 1.000 | 1.00 |
| Extremal deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.800±0.000 | 0.965 | 0.00 |
| Extremal deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.740±0.000 | 0.760 | 0.00 |
| Random deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.880±0.000 | 1.000 | 1.00 |
| Random deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.780±0.000 | 0.925 | 0.00 |
| Gaussian 0.064% | 0.516±0.092 | 0.515 | 0.02 | 0.020±0.000 | 0.000 | 0.00 |
| Gaussian 0.161% | 0.489±0.096 | 0.599 | 0.04 | 0.020±0.000 | 0.000 | 0.00 |
| Uniform noise 0.064% | 0.497±0.086 | 0.566 | 0.04 | 0.010±0.000 | 0.000 | 0.00 |
| Uniform noise 0.161% | 0.519±0.092 | 0.515 | 0.00 | 0.010±0.000 | 0.000 | 0.00 |
| Random insertion 30% | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Adversarial insertion 10% | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Shuffle | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Index rebuild | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Float16 | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| 8-bit quantization | 0.501±0.087 | 0.578 | 0.02 | 0.020±0.000 | 0.000 | 0.00 |
| 6-bit quantization | 0.494±0.086 | 0.587 | 0.04 | 0.000±0.000 | 0.000 | 0.00 |
| Transparency deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.928±0.009 | 1.000 | 1.00 |
| Transparency deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.718±0.013 | 0.605 | 0.00 |
| Modify30 (0.2) | 0.000±0.000 | 1.000 | 1.00 | 0.960±0.013 | 1.000 | 1.00 |

Table 27: DEEP: complete attack results for TVP and RS. Recovery is mean±standard deviation over 50 trials. TPR uses a nominal 1% FPR. Thresholds and achieved FPR appear in Table 24.

| | TVP | | | RS | | |
|---|---|---|---|---|---|---|
| Condition | BER | AUROC | TPR | BER | AUROC | TPR |
| No attack | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Rotation | 0.493±0.063 | 0.499 | 0.00 | 0.502±0.057 | 0.468 | 0.00 |
| Reflection | 0.502±0.062 | 0.463 | 0.00 | 0.497±0.075 | 0.483 | 0.02 |
| Axis permutation | 0.496±0.057 | 0.481 | 0.00 | 0.494±0.058 | 0.489 | 0.00 |
| Sign flips | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Translation | 0.496±0.059 | 0.488 | 0.00 | 0.497±0.058 | 0.481 | 0.00 |
| Uniform scaling | 0.512±0.061 | 0.415 | 0.00 | 0.515±0.059 | 0.403 | 0.00 |
| C1 | 0.505±0.056 | 0.455 | 0.00 | 0.505±0.054 | 0.453 | 0.00 |
| C2 | 0.503±0.068 | 0.466 | 0.00 | 0.503±0.064 | 0.464 | 0.00 |
| C3 | 0.499±0.060 | 0.489 | 0.00 | 0.502±0.056 | 0.462 | 0.00 |
| C4 | 0.504±0.064 | 0.468 | 0.00 | 0.507±0.063 | 0.429 | 0.02 |
| C5 | 0.502±0.065 | 0.462 | 0.00 | 0.500±0.071 | 0.487 | 0.00 |
| C6 | 0.493±0.063 | 0.499 | 0.00 | 0.502±0.057 | 0.468 | 0.00 |
| Shear 2% | 0.483±0.059 | 0.548 | 0.00 | 0.472±0.065 | 0.588 | 0.00 |
| Extremal deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Extremal deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Extremal deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Random deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Random deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Gaussian 0.064% | 0.503±0.073 | 0.475 | 0.00 | 0.504±0.059 | 0.452 | 0.00 |
| Gaussian 0.161% | 0.513±0.063 | 0.414 | 0.00 | 0.508±0.062 | 0.446 | 0.00 |
| Uniform noise 0.064% | 0.496±0.062 | 0.484 | 0.00 | 0.495±0.053 | 0.502 | 0.00 |
| Uniform noise 0.161% | 0.508±0.071 | 0.435 | 0.02 | 0.512±0.066 | 0.423 | 0.00 |
| Random insertion 30% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Adversarial insertion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Shuffle | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Index rebuild | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Float16 | 0.481±0.061 | 0.562 | 0.00 | 0.483±0.062 | 0.550 | 0.00 |
| 8-bit quantization | 0.516±0.068 | 0.415 | 0.00 | 0.516±0.066 | 0.414 | 0.00 |
| 6-bit quantization | 0.479±0.062 | 0.563 | 0.00 | 0.479±0.062 | 0.563 | 0.00 |
| Transparency deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Transparency deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Modify30 (0.2) | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |

Table 28: DEEP: complete attack results for SCPW and TabularMark. Recovery is mean±standard deviation over 50 trials. TPR uses a nominal 1% FPR. TabularMark reports MP. Thresholds and achieved FPR appear in Table 24.

| | SCPW | | | TabularMark | | |
|---|---|---|---|---|---|---|
| Condition | BER | AUROC | TPR | MP | AUROC | TPR |
| No attack | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Rotation | 0.480±0.083 | 0.572 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Reflection | 0.491±0.087 | 0.528 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Axis permutation | 0.439±0.071 | 0.715 | 0.02 | 0.000±0.000 | 0.000 | 0.00 |
| Sign flips | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Translation | 0.472±0.089 | 0.592 | 0.02 | 0.000±0.000 | 0.000 | 0.00 |
| Uniform scaling | 0.512±0.089 | 0.476 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C1 | 0.494±0.078 | 0.528 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C2 | 0.542±0.075 | 0.360 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C3 | 0.475±0.082 | 0.582 | 0.02 | 0.000±0.000 | 0.000 | 0.00 |
| C4 | 0.492±0.083 | 0.526 | 0.02 | 0.000±0.000 | 0.000 | 0.00 |
| C5 | 0.530±0.065 | 0.402 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C6 | 0.480±0.083 | 0.572 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Shear 2% | 0.494±0.086 | 0.520 | 0.00 | 0.190±0.000 | 0.000 | 0.00 |
| Extremal deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.910±0.000 | 1.000 | 1.00 |
| Extremal deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.850±0.000 | 0.990 | 1.00 |
| Extremal deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.580±0.000 | 0.000 | 0.00 |
| Random deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.880±0.000 | 1.000 | 1.00 |
| Random deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.780±0.000 | 0.925 | 0.00 |
| Gaussian 0.064% | 0.520±0.081 | 0.444 | 0.00 | 0.120±0.000 | 0.000 | 0.00 |
| Gaussian 0.161% | 0.496±0.077 | 0.522 | 0.00 | 0.020±0.000 | 0.000 | 0.00 |
| Uniform noise 0.064% | 0.486±0.086 | 0.537 | 0.04 | 0.090±0.000 | 0.000 | 0.00 |
| Uniform noise 0.161% | 0.506±0.077 | 0.491 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Random insertion 30% | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Adversarial insertion 10% | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Shuffle | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Index rebuild | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Float16 | 0.521±0.086 | 0.423 | 0.02 | 0.860±0.000 | 0.995 | 1.00 |
| 8-bit quantization | 0.478±0.083 | 0.578 | 0.02 | 0.000±0.000 | 0.000 | 0.00 |
| 6-bit quantization | 0.522±0.083 | 0.433 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Transparency deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.891±0.011 | 1.000 | 1.00 |
| Transparency deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.694±0.017 | 0.430 | 0.00 |
| Modify30 (0.2) | 0.000±0.000 | 1.000 | 1.00 | 0.923±0.018 | 1.000 | 1.00 |

Table 29: nomic: complete attack results for TVP and RS. Recovery is mean±standard deviation over 50 trials. TPR uses a nominal 1% FPR. Thresholds and achieved FPR appear in Table 24.

| | TVP | | | RS | | |
|---|---|---|---|---|---|---|
| Condition | BER | AUROC | TPR | BER | AUROC | TPR |
| No attack | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Rotation | 0.499±0.073 | 0.536 | 0.04 | 0.502±0.073 | 0.536 | 0.02 |
| Reflection | 0.506±0.066 | 0.487 | 0.00 | 0.504±0.060 | 0.520 | 0.00 |
| Axis permutation | 0.490±0.057 | 0.587 | 0.00 | 0.489±0.057 | 0.589 | 0.00 |
| Sign flips | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Translation | 0.506±0.063 | 0.507 | 0.00 | 0.505±0.063 | 0.513 | 0.00 |
| Uniform scaling | 0.503±0.062 | 0.511 | 0.00 | 0.504±0.064 | 0.505 | 0.00 |
| C1 | 0.509±0.067 | 0.498 | 0.02 | 0.505±0.060 | 0.500 | 0.00 |
| C2 | 0.499±0.063 | 0.543 | 0.00 | 0.502±0.054 | 0.528 | 0.00 |
| C3 | 0.491±0.060 | 0.579 | 0.00 | 0.474±0.065 | 0.636 | 0.00 |
| C4 | 0.488±0.056 | 0.592 | 0.00 | 0.472±0.060 | 0.648 | 0.00 |
| C5 | 0.506±0.065 | 0.525 | 0.00 | 0.508±0.057 | 0.504 | 0.00 |
| C6 | 0.499±0.073 | 0.536 | 0.04 | 0.502±0.073 | 0.536 | 0.02 |
| Shear 2% | 0.483±0.062 | 0.608 | 0.00 | 0.486±0.059 | 0.606 | 0.00 |
| Extremal deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Extremal deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Extremal deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Random deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Random deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Gaussian 0.064% | 0.499±0.065 | 0.532 | 0.02 | 0.511±0.056 | 0.473 | 0.00 |
| Gaussian 0.161% | 0.504±0.057 | 0.505 | 0.00 | 0.508±0.058 | 0.489 | 0.00 |
| Uniform noise 0.064% | 0.510±0.067 | 0.480 | 0.02 | 0.502±0.064 | 0.510 | 0.00 |
| Uniform noise 0.161% | 0.503±0.062 | 0.512 | 0.00 | 0.499±0.056 | 0.537 | 0.00 |
| Random insertion 30% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Adversarial insertion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Shuffle | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Index rebuild | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Float16 | 0.486±0.056 | 0.608 | 0.00 | 0.488±0.059 | 0.602 | 0.00 |
| 8-bit quantization | 0.510±0.067 | 0.490 | 0.00 | 0.496±0.070 | 0.559 | 0.02 |
| 6-bit quantization | 0.494±0.062 | 0.555 | 0.02 | 0.488±0.064 | 0.571 | 0.04 |
| Transparency deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Transparency deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Modify30 (0.2) | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |

Table 30: nomic: complete attack results for SCPW and TabularMark. Recovery is mean±standard deviation over 50 trials. TPR uses a nominal 1% FPR. TabularMark reports MP. Thresholds and achieved FPR appear in Table 24.

| | SCPW | | | TabularMark | | |
|---|---|---|---|---|---|---|
| Condition | BER | AUROC | TPR | MP | AUROC | TPR |
| No attack | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Rotation | 0.484±0.090 | 0.507 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Reflection | 0.494±0.091 | 0.478 | 0.02 | 0.000±0.000 | 0.000 | 0.00 |
| Axis permutation | 0.499±0.091 | 0.450 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Sign flips | 0.494±0.087 | 0.474 | 0.00 | 1.000±0.000 | 1.000 | 1.00 |
| Translation | 0.516±0.093 | 0.397 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Uniform scaling | 0.514±0.099 | 0.393 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C1 | 0.498±0.088 | 0.454 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C2 | 0.495±0.091 | 0.465 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C3 | 0.510±0.077 | 0.425 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C4 | 0.488±0.088 | 0.502 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C5 | 0.486±0.102 | 0.500 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C6 | 0.484±0.090 | 0.507 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Shear 2% | 0.532±0.089 | 0.352 | 0.00 | 0.210±0.000 | 0.000 | 0.00 |
| Extremal deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.840±0.000 | 0.990 | 1.00 |
| Extremal deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.740±0.000 | 0.760 | 0.00 |
| Extremal deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.600±0.000 | 0.000 | 0.00 |
| Random deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.880±0.000 | 1.000 | 1.00 |
| Random deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.780±0.000 | 0.925 | 0.00 |
| Gaussian 0.064% | 0.534±0.092 | 0.351 | 0.00 | 0.040±0.000 | 0.000 | 0.00 |
| Gaussian 0.161% | 0.483±0.097 | 0.515 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Uniform noise 0.064% | 0.494±0.083 | 0.468 | 0.00 | 0.060±0.000 | 0.000 | 0.00 |
| Uniform noise 0.161% | 0.488±0.088 | 0.480 | 0.02 | 0.000±0.000 | 0.000 | 0.00 |
| Random insertion 30% | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Adversarial insertion 10% | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Shuffle | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Index rebuild | 0.000±0.000 | 1.000 | 1.00 | 1.000±0.000 | 1.000 | 1.00 |
| Float16 | 0.518±0.072 | 0.378 | 0.00 | 0.960±0.000 | 1.000 | 1.00 |
| 8-bit quantization | 0.470±0.082 | 0.556 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| 6-bit quantization | 0.478±0.083 | 0.524 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Transparency deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.648±0.014 | 0.113 | 0.00 |
| Transparency deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.487±0.013 | 0.000 | 0.00 |
| Modify30 (0.2) | 0.000±0.000 | 1.000 | 1.00 | 0.969±0.020 | 1.000 | 1.00 |

Table 31: GIST: complete attack results for TVP and RS. Recovery is mean±standard deviation over 50 trials. TPR uses a nominal 1% FPR. Thresholds and achieved FPR appear in Table 24.

| | TVP | | | RS | | |
|---|---|---|---|---|---|---|
| Condition | BER | AUROC | TPR | BER | AUROC | TPR |
| No attack | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Rotation | 0.505±0.063 | 0.480 | 0.00 | 0.512±0.064 | 0.446 | 0.00 |
| Reflection | 0.513±0.066 | 0.457 | 0.00 | 0.511±0.064 | 0.473 | 0.00 |
| Axis permutation | 0.499±0.057 | 0.504 | 0.00 | 0.499±0.057 | 0.504 | 0.00 |
| Sign flips | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Translation | 0.512±0.058 | 0.439 | 0.00 | 0.512±0.059 | 0.441 | 0.00 |
| Uniform scaling | 0.502±0.060 | 0.495 | 0.00 | 0.502±0.060 | 0.497 | 0.00 |
| C1 | 0.505±0.061 | 0.484 | 0.00 | 0.504±0.058 | 0.484 | 0.02 |
| C2 | 0.499±0.069 | 0.490 | 0.00 | 0.495±0.065 | 0.525 | 0.00 |
| C3 | 0.495±0.059 | 0.534 | 0.00 | 0.493±0.057 | 0.551 | 0.00 |
| C4 | 0.488±0.057 | 0.560 | 0.00 | 0.498±0.058 | 0.509 | 0.00 |
| C5 | 0.494±0.068 | 0.521 | 0.02 | 0.498±0.065 | 0.512 | 0.00 |
| C6 | 0.505±0.063 | 0.480 | 0.00 | 0.512±0.064 | 0.446 | 0.00 |
| Shear 2% | 0.515±0.068 | 0.433 | 0.00 | 0.511±0.063 | 0.445 | 0.00 |
| Extremal deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Extremal deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Extremal deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Random deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Random deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Gaussian 0.064% | 0.497±0.068 | 0.518 | 0.00 | 0.504±0.066 | 0.506 | 0.00 |
| Gaussian 0.161% | 0.502±0.061 | 0.493 | 0.00 | 0.500±0.061 | 0.503 | 0.00 |
| Uniform noise 0.064% | 0.503±0.069 | 0.476 | 0.02 | 0.508±0.072 | 0.469 | 0.02 |
| Uniform noise 0.161% | 0.496±0.054 | 0.528 | 0.00 | 0.494±0.057 | 0.533 | 0.00 |
| Random insertion 30% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Adversarial insertion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Shuffle | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Index rebuild | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Float16 | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| 8-bit quantization | 0.484±0.067 | 0.567 | 0.02 | 0.484±0.067 | 0.569 | 0.00 |
| 6-bit quantization | 0.507±0.062 | 0.477 | 0.00 | 0.507±0.062 | 0.477 | 0.00 |
| Transparency deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Transparency deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |
| Modify30 (0.2) | 0.000±0.000 | 1.000 | 1.00 | 0.000±0.000 | 1.000 | 1.00 |

Table 32: GIST: complete attack results for SCPW and TabularMark. Recovery is mean±standard deviation over 50 trials. TPR uses a nominal 1% FPR. TabularMark reports MP; GIST identifier collisions are retained. Thresholds and achieved FPR appear in Table 24.

| | SCPW | | | TabularMark | | |
|---|---|---|---|---|---|---|
| Condition | BER | AUROC | TPR | MP | AUROC | TPR |
| No attack | 0.000±0.000 | 1.000 | 1.00 | 1.010±0.000 | 1.000 | 1.00 |
| Rotation | 0.524±0.076 | 0.365 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Reflection | 0.526±0.089 | 0.356 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Axis permutation | 0.474±0.074 | 0.545 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Sign flips | 0.324±0.067 | 0.927 | 0.04 | 1.010±0.000 | 1.000 | 1.00 |
| Translation | 0.503±0.093 | 0.456 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Uniform scaling | 0.484±0.096 | 0.512 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C1 | 0.497±0.083 | 0.460 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C2 | 0.495±0.089 | 0.448 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C3 | 0.476±0.077 | 0.519 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C4 | 0.469±0.073 | 0.550 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C5 | 0.496±0.082 | 0.466 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| C6 | 0.524±0.076 | 0.365 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Shear 2% | 0.522±0.088 | 0.383 | 0.00 | 0.170±0.000 | 0.000 | 0.00 |
| Extremal deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.940±0.000 | 1.000 | 1.00 |
| Extremal deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.860±0.000 | 0.990 | 1.00 |
| Extremal deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.680±0.000 | 0.295 | 0.00 |
| Random deletion 10% | 0.000±0.000 | 1.000 | 1.00 | 0.890±0.000 | 1.000 | 1.00 |
| Random deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.790±0.000 | 0.925 | 0.00 |
| Gaussian 0.064% | 0.494±0.083 | 0.456 | 0.00 | 0.080±0.000 | 0.000 | 0.00 |
| Gaussian 0.161% | 0.478±0.098 | 0.534 | 0.00 | 0.010±0.000 | 0.000 | 0.00 |
| Uniform noise 0.064% | 0.514±0.095 | 0.401 | 0.00 | 0.070±0.000 | 0.000 | 0.00 |
| Uniform noise 0.161% | 0.517±0.104 | 0.399 | 0.02 | 0.010±0.000 | 0.000 | 0.00 |
| Random insertion 30% | 0.000±0.000 | 1.000 | 1.00 | 1.010±0.000 | 1.000 | 1.00 |
| Adversarial insertion 10% | 0.000±0.000 | 1.000 | 1.00 | 1.010±0.000 | 1.000 | 1.00 |
| Shuffle | 0.000±0.000 | 1.000 | 1.00 | 1.010±0.000 | 1.000 | 1.00 |
| Index rebuild | 0.000±0.000 | 1.000 | 1.00 | 1.010±0.000 | 1.000 | 1.00 |
| Float16 | 0.007±0.013 | 1.000 | 1.00 | 1.010±0.000 | 1.000 | 1.00 |
| 8-bit quantization | 0.506±0.085 | 0.419 | 0.00 | 0.020±0.000 | 0.000 | 0.00 |
| 6-bit quantization | 0.457±0.073 | 0.597 | 0.00 | 0.000±0.000 | 0.000 | 0.00 |
| Transparency deletion 20% | 0.000±0.000 | 1.000 | 1.00 | 0.893±0.013 | 1.000 | 1.00 |
| Transparency deletion 40% | 0.000±0.000 | 1.000 | 1.00 | 0.737±0.006 | 0.694 | 0.00 |
| Modify30 (0.2) | 0.000±0.000 | 1.000 | 1.00 | 1.001±0.010 | 1.000 | 1.00 |