WATERMARK ANY IMAGE: A UNIFIED FRAMEWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Embedding data in images while maintaining visual quality and ensuring reliable detection and decoding remains a challenge in image watermarking. We introduce UniMark, a unified watermark framework for zero-bit detection, multi-bit detection, and multi-bit decoding that operates directly on existing images without accessing or modifying the generator. UniMark calibrates projection scores in reconstructable image representations using unwatermarked distributions and modifies these scores to embed watermark information. We develop statistical watermark constructions and corresponding likelihood-ratio tests for detection and decoding. These tests are optimal under independence, while finite-sample analysis demonstrates near-optimal performance. Experiments on three real-image datasets and images from four generative models demonstrate high image fidelity at near-perfect detection rates across diverse image sources. UniMark supports payloads of up to 32,768 bits, twice the largest payload of the existing largest payload watermark method, and achieves a 92% full-message recovery rate at 8,192 bits under clean conditions, which existing methods have not demonstrated.

1 INTRODUCTION

As AI-generated images become more common, tracing and verifying their sources becomes increasingly important (Fernandez et al., 2023; Wen et al., 2023; Yang et al., 2024). Image watermarking provides a practical approach by embedding imperceptible signals into images and subsequently detecting them (Zhu et al., 2018; Bui et al., 2025). It includes three tasks:

- **Zero-bit detection**: determines whether a watermark is present without an embedded message.
- **Multi-bit detection**: determines whether a message-bearing watermark is present.
- **Multi-bit decoding**: recovers the embedded message.

The key challenge is to make watermarks reliably detectable while keeping changes to the image visually imperceptible. Embedding more information supports finer-grained source identification and tracing (Tancik et al., 2020; Yu et al., 2021; Kim et al., 2024). Ideally, watermarking should also work across different generation methods. However, embedding more information or making watermarks easier to detect may require larger image changes, compromising visual quality. Methods tailored to a particular generator may also be difficult to extend to other generation methods.

Existing image watermarking methods can be grouped into post-hoc and generator-integrated approaches. Post-hoc methods embed watermarks into existing images without accessing the generator. Training-free approaches include hand-crafted methods such as DWT-DCT (ShieldMnt, 2021), which modify transform coefficients, and FNNS (Kishore et al., 2022), which optimizes image pixels while keeping a decoding network fixed. Methods using learned representations follow different embedding strategies: SSL (Fernandez et al., 2022) optimizes images to encode watermarks in a pretrained self-supervised feature space, whereas TrustMark (Bui et al., 2025) trains an embedder and extractor to insert and recover messages. These approaches support generator-independent deployment, with different requirements for network training and per-image optimization.

Generator-integrated methods embed watermarks through the generator's parameters, latent representations, or sampling process. For diffusion models, Stable Signature (Fernandez et al., 2023) and LaWa (Rezaei et al., 2024) embed watermarks through the decoder, while Gaussian Shading (Yang et al., 2024) and MaxMark (Chang et al., 2026) encode messages in latent representations. For GANs, PTW (Lukas & Kerschbaum, 2023) fine-tunes the generator. For autoregressive models,

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

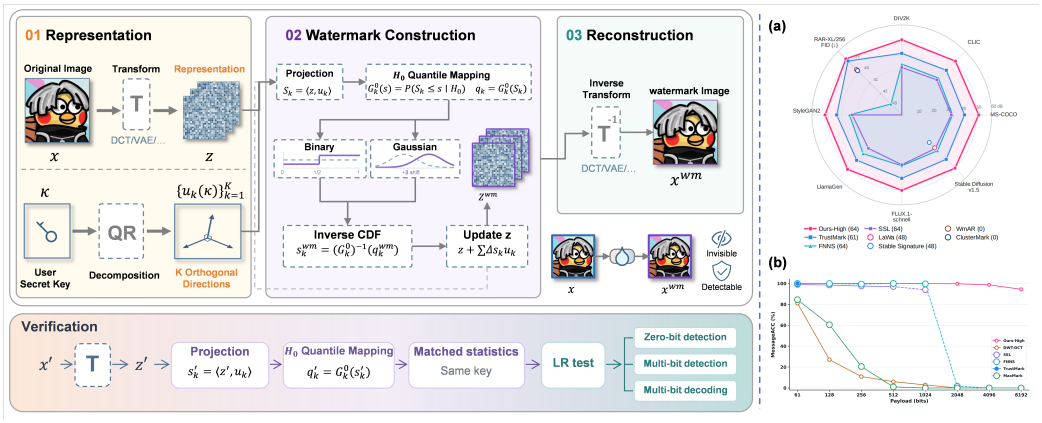

Figure 1: **Overview of UniMark and evaluation.** Left: UniMark framework for watermark embedding and verification. Right: (a) Image quality across different image sources. (b) Full-message recovery accuracy across different payloads. Solid segments indicate mean PSNR $\geq$ 40 dB at both endpoints, while dashed segments indicate that at least one endpoint is below 40 dB; marker size indicates the total injection and extraction time over 200 images.

WmAR (Jovanović et al., 2025) and ClusterMark (Lukovnikov et al., 2026) modify visual-token sampling. These approaches depend on components of the underlying generation pipeline. Further discussion of related work is provided in Appendix A.

In this paper, we introduce **UniMark**, a unified framework for watermarking any image that supports zero-bit detection, multi-bit detection, and multi-bit decoding. UniMark embeds watermark information through a unified statistical construction and uses likelihood-ratio tests for detection and decoding. An invertible or approximately invertible transform maps unwatermarked images into a representation space. We then project these representations along selected directions and estimate the score distributions under the unwatermarked hypothesis. To embed a watermark, we modify the scores according to our statistical construction, then reconstruct the watermarked image using the inverse transform. Watermark detection and message decoding are performed using likelihood-ratio tests. The framework supports different invertible image transforms and distributional constructions, and operates directly on existing images without accessing or modifying the generator (Table 5). The overall framework and representative evaluation results are shown in Figure 1, with detailed evaluation settings provided in Appendix C.1.

We evaluate UniMark on three real-image datasets and images from four generative models across all three tasks. At near-perfect detection rates, UniMark achieves higher image quality than the compared baselines. It supports payloads of up to 32,768 bits while maintaining high image quality, doubling the largest payload supported by existing watermarking methods (Chang et al., 2026). It also achieves the highest processing efficiency among the compared methods. Beyond bit-level accuracy, UniMark achieves a 92% full-message recovery rate for 8,192-bit payloads under clean conditions. To our knowledge, no prior method has demonstrated reliable full-message recovery at this payload size. Our contributions are as follows:

- We introduce UniMark, a unified statistical framework for zero-bit detection, multi-bit detection, and multi-bit decoding.
- We develop Binary and Gaussian constructions, establish oracle performance upper bounds, and demonstrate near-optimal performance through finite-sample analysis.
- Extensive experiments across all three tasks demonstrate the best image quality among baselines at high detection rates, with large payload capacity, high message recovery rates, and high processing efficiency.

2 METHOD AND THEORETICAL ANALYSIS

2.1 PROBLEM SETUP & NOTATIONS

Image watermarking generally consists of two processes: embedding and verification. During embedding, given an input image x and a secret key κ, the watermarking algorithm produces a watermarked image x^{wm}; in the multi-bit setting, it additionally takes a message m. During verification, the corresponding detection algorithm determines whether the watermark is present from the observed image and the secret key κ, and recovers the embedded message in the multi-bit setting.

For the three watermarking tasks introduced in Section 1, let X denote the target image. Let P_0, P_{zero}, and P_m denote the image distributions without a watermark, with a zero-bit watermark, and with an embedded message m, respectively. All watermarked distributions are defined for a fixed secret key κ. Then three tasks can be formulated as follows:

- **Zero-bit detection:** $H_0 : X \sim P_0$ vs. $H_1 : X \sim P_{\mathrm{zero}}$.
- **Multi-bit detection:** $H_0 : X \sim P_0$ vs. $H_1 : X \sim P_m$ for $m \in \{0,1\}^B$.
- **Multi-bit decoding:** $H_{b,0} : m_b = 0$ vs. $H_{b,1} : m_b = 1$ for $b = 1, \ldots, B$.

In the multi-bit setting, $m = (m_1, \ldots, m_B) \in \{0,1\}^B$ denotes a B-bit message, where multi-bit detection determines the presence of a message-bearing watermark and multi-bit decoding further recovers its individual bits.

In this paper, we embed watermarks in a representation space $\mathcal{Z}$. Specifically, a transform $T : \mathcal{X} \to \mathcal{Z}$ maps an input image x to its representation $z = T(x)$. The representation z is assumed to support image reconstruction. Here $T(\cdot)$ can be a conventional transform such as the discrete cosine transform (DCT) (Ahmed et al., 1974) or discrete Fourier transform (DFT) (Winograd, 1978), or a learned transform such as a variational autoencoder (VAE) (Kingma & Welling, 2014) that produces a reconstructable latent representation.

Notations. We use $\mathbb{P}(\cdot)$ to denote the probability of an event and $p(\cdot)$ to denote a probability mass or density function, as appropriate. The indicator $\mathbb{I}(A)$ equals 1 if A holds and 0 otherwise. The notation $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product, and $\| \cdot \|_2$ denotes the Euclidean norm. We write $X \sim D$ to indicate that X follows distribution D. We denote the uniform, Gaussian, and Bernoulli distributions by $\mathcal{U}(0,1)$, $\mathcal{N}(\mu, \sigma^2)$, and $\mathrm{Bernoulli}(\pi)$, respectively, where μ and σ^2 are the Gaussian mean and variance, and π is the Bernoulli success probability. We use $\Lambda\mathcal{B}$ and $\Lambda\mathcal{G}$ to denote the likelihood ratio statistics for the Binary and Gaussian constructions, respectively. The superscripts zero, $\mathrm{multi,det}$, and $\mathrm{multi,dec}$ indicate zero-bit detection, multi-bit watermark detection, and multi-bit message decoding, respectively. A subscript b is used when the statistic is associated with the b-th message bit.

2.2 WATERMARK CONSTRUCTION

2.2.1 STATISTICAL PROJECTION AND CALIBRATION

For the hypothesis testing task, let $\mathbf{Y} = (Y_1, \ldots, Y_K)$ denote K statistical observations obtained from the image. In the general case, the likelihood ratio for distinguishing H_1 from H_0 is defined by the joint distributions of these observations:

$$\Lambda^*(\mathbf{Y}) = \frac{p(\mathbf{Y} \mid H_1)}{p(\mathbf{Y} \mid H_0)}. \tag{1}$$

By the Neyman–Pearson lemma (Neyman & Pearson, 1933), thresholding this likelihood ratio yields the most powerful test at a fixed false-positive rate. However, directly computing this likelihood ratio requires modeling the K-dimensional joint distributions of these observations under H_0 and H_1, which are intractable.

If the observations are independent under each hypothesis, the joint distribution factorizes into the product of K one-dimensional marginal distributions, reducing the K-dimensional modeling problem to K one-dimensional ones. The resulting log-likelihood ratio is

$$\log \Lambda_{\mathrm{ind}}(\mathbf{Y}) = \sum_{k=1}^{K} \log \frac{p(Y_k \mid H_1)}{p(Y_k \mid H_0)}. \tag{2}$$

In practice, the observations are not guaranteed to be strictly independent. To characterize this dependence, we define the dependence factor

$$d_h(\mathbf{Y}) = \frac{p(\mathbf{Y} \mid H_h)}{\prod_{k=1}^{K} p(Y_k \mid H_h)}. \tag{3}$$

The Neyman–Pearson optimal log-likelihood ratio statistic can be decomposed as

$$\log \Lambda^*(\mathbf{Y}) = \sum_{k=1}^{K} \log \frac{p(Y_k \mid H_1)}{p(Y_k \mid H_0)} + \log \frac{d_1(\mathbf{Y})}{d_0(\mathbf{Y})} = \log \Lambda_{\mathrm{ind}}(\mathbf{Y}) + \Delta(\mathbf{Y}). \tag{4}$$

where $\Delta(\mathbf{Y}) = \log[d_1(\mathbf{Y})/d_0(\mathbf{Y})]$ denotes the correction to the full log-likelihood ratio induced by dependence among the observations. Computing $\Delta(\mathbf{Y})$ directly requires the high-dimensional joint distributions of the observations under H_0 and H_1, which are unknown in practice. Therefore, the true joint likelihood ratio $\Lambda^*(\mathbf{Y})$ is not directly computable in practice, and we omit $\Delta(\mathbf{Y})$ and use the independence-form likelihood ratio $\Lambda_{\mathrm{ind}}(\mathbf{Y})$ as the practical statistic.

To reduce the effect of dependence, given a secret key κ, we generate K key-dependent orthonormal directions $\mathcal{U}_\kappa = \{u_k(\kappa)\}_{k=1}^{K}$ in the representation space $\mathcal{Z}$. In practice, these directions are efficiently constructed using a key-dependent coordinate permutation and 256×256 orthogonal blocks, as detailed in Appendix C.3. These directions satisfy $u_i^\top u_j = 0$ for $i \neq j$ and $\|u_k\|_2 = 1$, which helps reduce dependence among the resulting statistical observations.

For each direction u_k, the projection score is defined as $S_k = \langle z, u_k \rangle$. Its cumulative distribution function under H_0, $G_k^0(s) = \mathbb{P}(S_k \leq s \mid H_0)$, is estimated from the corresponding projection scores of unwatermarked calibration images. We then map each projection score to its quantile under H_0 as $q_k = G_k^0(S_k)$. With accurate calibration, $q_k \sim \mathcal{U}(0,1)$ under H_0. This calibration maps projection scores from different directions into a shared quantile space for watermark construction.

In the shared quantile space, we define two watermark constructions: Binary and Gaussian. Both constructions share the same projection, calibration, and reconstruction pipeline, while using different watermark transformations in the shared quantile space; the Gaussian formulation is provided in Appendix B.1.

2.2.2 Binary Construction

Using 0.5 as the boundary, the calibrated quantile q_k is divided into two regions, with the binary statistic defined as $Y_k = \mathbb{I}(q_k > 1/2)$. Since $q_k \sim \mathcal{U}(0,1)$ under H_0, the two regions have equal probability, giving $Y_k \mid H_0 \sim \mathrm{Bernoulli}(1/2)$.

Watermark embedding adds a controlled bias toward one of the two regions, referred to as the target region, with the other being the non-target region. By increasing the probability that a quantile falls in the target region above its unwatermarked level of $1/2$, the construction creates a statistical signal that can be detected by aggregating evidence across statistics. To control the strength of this bias, we introduce a strength parameter $\beta > 0$ and set the probabilities of the target and non-target regions to $p_\beta = \frac{e^\beta}{1+e^\beta}$ and $\rho_\beta = \frac{1}{1+e^\beta}$, respectively, where $p_\beta + \rho_\beta = 1$ and $p_\beta > 1/2$. Larger values of β assign more probability mass to the target region.

When the right half is the target region, we remap the calibrated quantile as $q_k^{\mathrm{wm}} = \frac{q_k}{2\rho_\beta}$ for $q_k \leq \rho_\beta$, and $q_k^{\mathrm{wm}} = \frac{1}{2} + \frac{q_k - \rho_\beta}{2p_\beta}$ for $q_k > \rho_\beta$. This mapping gives $P(q_k^{\mathrm{wm}} > 1/2) = p_\beta$ and $P(q_k^{\mathrm{wm}} \leq 1/2) = \rho_\beta$. When the left half is the target region, we use the symmetric mapping around 0.5.

For zero-bit watermarking, the right half is fixed as the target region, giving $P(Y_k = 1 \mid H_1) = p_\beta$ and $P(Y_k = 0 \mid H_1) = \rho_\beta$. Together with $Y_k \sim \mathrm{Bernoulli}(1/2)$ under H_0, the independence-form log-likelihood ratio is

$$\log \Lambda\mathcal{B}^{\mathrm{zero}}(\mathbf{Y}) = C \log \frac{p_\beta}{\rho_\beta} + K \log(2\rho_\beta), \qquad C = \sum_{k=1}^{K} Y_k. \tag{5}$$

For multi-bit watermarking, we partition the K statistic indices into B disjoint sets $\{\mathcal{I}_b\}_{b=1}^{B}$, each containing M indices and assigned to one message bit m_b. For $k \in \mathcal{I}_b$, the right half ($q_k > 1/2$)

is selected as the target region when $m_b = 0$, and the left half ($q_k \leq 1/2$) when $m_b = 1$, giving $P(Y_k = 1 \mid m_b = 0) = P(Y_k = 0 \mid m_b = 1) = p_\beta$. Under independence, the log-likelihood ratio between $m_b = 1$ and $m_b = 0$ for the b-th message bit is

$$\log \Lambda_b^{\mathrm{multi,dec}} = \sum_{k \in \mathcal{I}_b} \log \frac{p(Y_k \mid m_b = 1)}{p(Y_k \mid m_b = 0)} = \beta(M - 2C_b), \qquad C_b = \sum_{k \in \mathcal{I}_b} Y_k. \tag{6}$$

Thus, $\hat{m}_b = 0$ when $C_b \geq M/2$, and $\hat{m}_b = 1$ when $C_b < M/2$. Detailed likelihood derivations for multi-bit detection and decoding, together with their optimality proofs, are provided in Appendix B.2.1. The corresponding detection, decoding, and optimality analyses for the Gaussian construction are provided in Appendix B.2.2.

2.2.3 Watermark Injection and Reconstruction

After obtaining the target quantile q_k^{wm} from the Binary construction or the Gaussian construction in Appendix B.1, we map it back through the inverse H_0 calibration as $S_k^{\mathrm{wm}} = (G_k^0)^{-1}(q_k^{\mathrm{wm}})$. The image representation is then updated along the K orthonormal directions according to the changes in the projection scores:

$$z^{\mathrm{wm}} = z + \sum_{k=1}^{K} (S_k^{\mathrm{wm}} - S_k)\, u_k. \tag{7}$$

Since the directions are orthonormal, the updated representation satisfies $\langle z^{\mathrm{wm}}, u_k \rangle = S_k^{\mathrm{wm}}$ for every k, while preserving the component orthogonal to these directions. Finally, the modified representation is mapped back to the image space through the corresponding reconstruction transform, giving $x^{\mathrm{wm}} = T^{-1}(z^{\mathrm{wm}})$.

2.3 Optimality and Dependence Analysis

Under independence, the proposed statistic in Eq. 4 equals the true joint likelihood ratio and thus yields a most powerful test at a fixed false-positive rate by the Neyman–Pearson lemma. For a target false-positive rate α, let G_α denote the detection-power gap between the practical detector and the Neyman–Pearson optimal detector. When the practical detector satisfies $\mathrm{FPR}_\phi \leq \alpha$, the Neyman–Pearson lemma ensures that its detection power does not exceed the optimum, giving $0 \leq G_\alpha \leq \mathrm{FNR}_\phi$, where $\mathrm{FNR}_\phi = P_{H_1}(\phi(\mathbf{Y}) = 0)$ denotes the false-negative rate of the practical detector under H_1. Thus, under the target false-positive rate constraint, a small false-negative rate directly guarantees that the practical detection power is close to the optimal detection power.

For multi-bit decoding, let G_{bit} denote the bit-accuracy gap between the practical decoder and the optimal bit decoder under the true joint distribution. The gap satisfies $0 \leq G_{\mathrm{bit}} \leq \mathrm{BER}$, where BER denotes the bit error rate of the practical decoder. Thus, a small bit error rate directly guarantees that the bit accuracy of the practical decoder is close to that of the optimal decoder. The optimality gaps under dependence and the corresponding finite-sample guarantees are provided in Appendix B.3.

3 Experiments

3.1 Experimental Setup

Datasets and Image Sources. We evaluate our method on both real and AI-generated images. Cross-source and post-hoc evaluations use a resolution of 512×512, while generator-specific comparisons follow the native resolution settings of the corresponding baselines. For real images, we use 500 images from DIV2K (Agustsson & Timofte, 2017), 102 images from CLIC2020 (Toderici et al., 2020), and 2,000 images from MS-COCO (Lin et al., 2014). For AI-generated images, we use Stable Diffusion v1.5 (Rombach et al., 2022), FLUX.1-schnell (Black Forest Labs, 2024), LlamaGen (Sun et al., 2024), and StyleGAN2 (Karras et al., 2020), with 200 generated images from each model. The resulting test set contains 3,402 images and covers real images together with different generative paradigms, including diffusion, autoregressive, and GAN models.

Baselines. We group the baselines into post-hoc watermarking methods and watermarking methods for specific generators. For post-hoc watermarking, we compare with DWT-DCT (ShieldMnt, 2021), SSL (Fernandez et al., 2022), FNNS (Kishore et al., 2022), and TrustMark (Bui et al., 2025), which embed watermarks into images after generation. For generator-integrated watermarking, we consider representative methods for diffusion and autoregressive generation, including Stable Signature (Fernandez et al., 2023), LaWa (Rezaei et al., 2024), WmAR (Jovanović et al., 2025), and ClusterMark (Lukovnikov et al., 2026). For high-capacity multi-bit watermarking, we further compare with MaxMark (Chang et al., 2026) as a baseline for long-payload embedding. These baselines cover different watermarking paradigms and allow us to compare our method with existing approaches across different settings.

Implementation Details. In our experiments, we instantiate the representation transform T as the discrete cosine transform (DCT) (Ahmed et al., 1974) and perform watermark embedding and detection in the resulting DCT representation space. We further validate the generalization of the framework across different reconstructable representations in Appendix D.1. Based on the selected frequency subspace, we use two configurations, Ours-Mid and Ours-High, which embed watermarks in the middle- and high-frequency subspaces, respectively. In the main experiments, we use the Gaussian construction for zero-bit watermarking and the Binary construction for multi-bit watermarking. Both constructions are further evaluated across different image sources and representation spaces in the appendix. When comparing with existing methods, we adjust the embedded message length according to the payload used by the corresponding baseline to ensure a consistent comparison. We further evaluate multi-bit watermarking over a wider range of payload sizes to study detection, decoding, and image quality as the message length increases.

Additional experimental settings, implementation details, and evaluation protocols are provided in Appendix C.

3.2 Main Results

As shown in Table 1, Ours-Mid and Ours-High achieve 100% TPR across all image sources and maintain nearly perfect BitAcc and MessageAcc, showing stable detection and decoding across different image sources. Ours-High consistently improves image quality with negligible loss in message accuracy. Results show that our method further reduces watermark embedding distortion when detection and decoding performance are already near saturation. For zero-bit watermarking, both constructions also outperform the baselines, and detailed zero-bit results are provided in Appendix D.5.

As shown in Table 2, our method maintains reliable watermarking performance on diffusion and autoregressive models without accessing or modifying the generation process. It also preserves higher image quality than the baselines at near-perfect detection or decoding rates.

3.3 Further Results

High-Capacity Watermarking. Table 3 shows how multi-bit watermarking scales to high-capacity payloads. As the payload increases from $8{,}192$ to $32{,}768$ bits, Ours-High maintains 100% TPR, while BitAcc remains high, showing stable watermark detection and bit-wise decoding at high payloads. In contrast, MessageAcc decreases from 92.0% to 25.5%, since a message is recovered correctly only when all of its bits are decoded correctly, making small bit-wise errors more influential as the message length increases. At $8{,}192$ and $16{,}384$ bits, our method achieves higher BitAcc than MaxMark and reaches MessageAcc values of 92.0% and 67.7%, respectively, while MaxMark obtains 0% MessageAcc in both settings. Across these payloads, Ours-High maintains the highest PSNR and SSIM, showing that increasing the payload mainly affects message recovery while image quality remains stable. The near-zero ΔFID at $8{,}192$ and $16{,}384$ bits further indicates a small effect of high-capacity watermarking on the generated image distribution. Further analysis shows that complete recovery of long messages also depends on image content, with images containing fewer high-frequency structures generally supporting more reliable error-free recovery, as shown in Appendix D.2.

Dataset	Method (bits)	Detection and Decoding			Image Quality	
		TPR@0.1% ↑	BitAcc ↑	MessageAcc ↑	PSNR ↑	SSIM ↑
DIV2K	DWT-DCT (48)	57.80%	85.68%	25.80%	36.16	0.9656
	SSL (64)	98.00%	99.89%	97.60%	33.08	0.9184
	FNNS (64)	99.60%	100.00%	99.80%	33.91	0.9515
	TrustMark (61)	99.80%	99.98%	99.80%	41.08	0.9909
	Ours-Mid (64)	100.00%	100.00%	100.00%	47.75	0.9935
	Ours-High (64)	100.00%	99.99%	99.40%	**50.11**	**0.9963**
CLIC	DWT-DCT (48)	54.90%	85.74%	33.33%	38.26	0.9667
	SSL (64)	100.00%	99.97%	98.04%	33.09	0.8934
	FNNS (64)	100.00%	100.00%	100.00%	34.71	0.9479
	TrustMark (61)	100.00%	100.00%	100.00%	42.12	0.9910
	Ours-Mid (64)	100.00%	100.00%	100.00%	48.07	0.9927
	Ours-High (64)	100.00%	100.00%	100.00%	**50.70**	**0.9961**
MS-COCO	DWT-DCT (48)	64.55%	88.73%	35.55%	37.31	0.9666
	SSL (64)	99.35%	99.98%	99.30%	33.10	0.8998
	FNNS (64)	99.25%	99.99%	99.70%	33.91	0.9414
	TrustMark (61)	100.00%	100.00%	100.00%	41.89	0.9911
	Ours-Mid (64)	100.00%	100.00%	100.00%	47.86	0.9931
	Ours-High (64)	100.00%	100.00%	99.90%	**51.42**	**0.9970**
AI-generated	DWT-DCT (48)	51.75%	83.91%	24.38%	38.41	0.9623
	SSL (64)	98.50%	99.94%	98.25%	33.10	0.8815
	FNNS (64)	99.63%	100.00%	100.00%	34.81	0.9415
	TrustMark (61)	100.00%	100.00%	100.00%	42.59	0.9906
	Ours-Mid (64)	100.00%	100.00%	100.00%	48.16	0.9919
	Ours-High (64)	100.00%	100.00%	100.00%	**50.79**	**0.9956**
Macro Avg.	DWT-DCT (48)	57.25%	86.02%	29.76%	37.54	0.9653
	SSL (64)	98.96%	99.94%	98.30%	33.09	0.8983
	FNNS (64)	99.62%	100.00%	99.88%	34.33	0.9456
	TrustMark (61)	99.95%	100.00%	99.95%	41.92	0.9909
	Ours-Mid (64)	100.00%	100.00%	100.00%	47.96	0.9928
	Ours-High (64)	100.00%	100.00%	99.83%	**50.75**	**0.9963**

Table 1: Comparison with post-hoc multi-bit watermarking methods across different image sources. Higher values are better for all metrics. The payload size of each method is shown in parentheses. For image-quality metrics, the best value is shown in **bold**, and the second-best value is underlined.

(a) Diffusion-based methods

Method	Detection and Decoding			Image Quality	
	TPR@0.1% ↑	BitAcc ↑	MessageAcc ↑	PSNR ↑	SSIM ↑
Stable Signature	99.93%	99.26%	81.05%	29.28	0.8887
LaWa	100.00%	100.00%	100.00%	29.32	0.8608
Ours-Mid	100.00%	100.00%	100.00%	48.13	0.9931
Ours-High	100.00%	100.00%	100.00%	**50.71**	**0.9963**

(b) Autoregressive methods

Method	Detection	Image Quality		
	TPR@0.1% ↑	PSNR ↑	SSIM ↑	ΔFID ↓
ClusterMark	100.00%	–	–	0.8684
WmAR	100.00%	–	–	1.5371
Ours-Mid	100.00%	50.07	0.9962	**-0.0090**
Ours-High	100.00%	**56.96**	**0.9992**	-0.0049

Table 2: Comparison with generator-integrated watermarking methods across different generation paradigms. Panel (a) compares diffusion-based methods, while panel (b) compares autoregressive methods. TPR is reported at a false-positive rate of 0.1%. Higher values are better for TPR, BitAcc, MessageAcc, PSNR, and SSIM, while lower values are better for ΔFID. For image-quality metrics, the best value is shown in **bold**, and the second-best value is underlined.

Robustness. We evaluate watermark robustness under JPEG compression, resizing, Gaussian blur, brightness changes, Gaussian noise, contrast changes, and combined attacks. As shown in Table 4(b), our method maintains strong detection and decoding performance under perturbations in both post-hoc and diffusion settings. In the autoregressive setting, our method achieves an average TPR of 98.39% under the stricter FPR of 0.1%, exceeding ClusterMark and WmAR. For high-capacity watermarking, the average TPR@1% remains at 100% as the payload increases from $8{,}192$ to $32{,}768$ bits, while BitAcc decreases from 96.65% to 90.40%, indicating that larger payloads mainly affect message decoding under perturbations. Overall, watermark presence detection remains stable across different settings and image perturbations, while message decoding is more sensitive to payload size and perturbations. Detailed per-attack results are provided in Appendix D.4.

Payload	Method	Detection and Decoding			Image Quality		
		TPR@0.1% ↑	BitAcc ↑	MessageAcc ↑	PSNR ↑	SSIM ↑	ΔFID ↓
8,192	MaxMark	–	96.2883%	0.0%	–	–	+2.5216
	Ours-High	**100.0%**	**99.9130%**	**92.0%**	42.1033	0.969608	**-0.0125**
16,384	MaxMark	–	94.8791%	0.0%	–	–	+2.8860
	Ours-High	**100.0%**	**99.7131%**	**67.7%**	42.1795	0.970003	**-0.0143**
32,768	MaxMark	–	–	–	–	–	–
	Ours-High	**100.0%**	**99.2359%**	**25.5%**	42.1795	0.970004	–

Table 3: High-capacity multi-bit watermarking comparison with MaxMark. Experiments are conducted on 1,000 images generated by Stable Diffusion v1.5. For ΔFID, all methods use the same 1,000 COCO captions and the corresponding real COCO images as the common reference set. Unavailable results are denoted by "–".

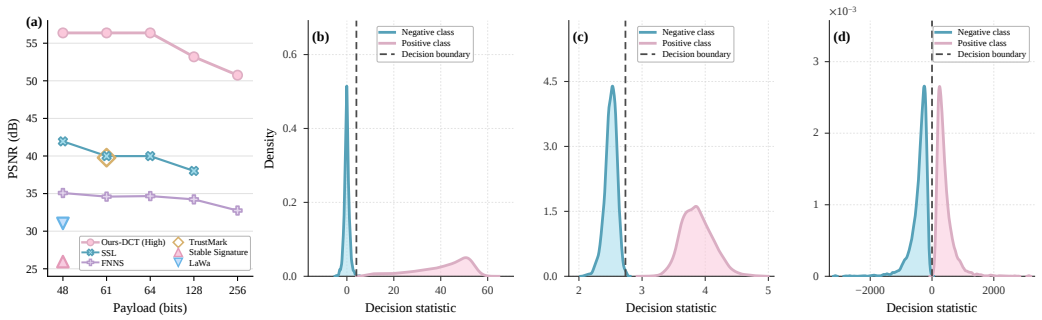

Figure 2: Image quality under predefined performance requirements and decision statistic distributions. (a) Image quality of different methods under TPR $\geq 99\%$ for zero-bit detection and BitAcc $\geq 99\%$ with MessageAcc $> 90\%$ for multi-bit watermarking. (b)–(d) Decision statistic distributions for zero-bit detection, multi-bit detection, and bit-wise decoding, respectively. Dashed lines indicate the corresponding decision boundaries.

Statistic Distributions. Figure 2(b) shows that zero-bit watermarking shifts the detection statistic under H_1 toward larger values, producing clear separation from H_0. Figure 2(c) shows a similar separation between unwatermarked and watermarked samples for multi-bit watermark detection. In Figure 2(d), the decoding statistics for bit 0 and bit 1 lie on opposite sides of zero, consistent with the bit-wise decision rule defined by the sign of the likelihood ratio. These results show that the constructed statistics support both watermark presence detection and message decoding.

Image Quality and Qualitative Results. Figure 2(a) compares image quality at operating points selected according to the protocol in Appendix C.1. Our method achieves higher PSNR at all comparable payloads and remains above 50 dB at 256 bits. Figure 3(a) shows the visual results under different watermark settings. From zero-bit to 32,768-bit watermarking, the watermarked images remain visually close to the original. All displayed watermarks are successfully detected, with complete message recovery in all cases except for a one-bit error in the 32,768 case. Figure 3(b) shows 64-bit results across different image sources, with an average PSNR of 48.039 dB over the six examples, indicating high image quality across different image contents. We further compare the $20\times$ amplified residuals of our method with those of other methods in Appendix D.3 to visualize the image modifications introduced by different watermarking methods.

Efficiency. As shown in Table 4(a), Ours-DCT processes 200 Stable Diffusion v1.5 images in 1.298 s, $3.18\times$ faster than the fastest baseline, while achieving 100% TPR, BitAcc, and MessageAcc. Compared with TrustMark at the same detection and decoding performance, our method reduces the total time from 4.129 s to 1.298 s while increasing PSNR from 39.660 dB to 55.370 dB.

(a) Efficiency

Method	Bits	Time (s)			TPR / Bit / Msg	PSNR
		Inject	Detect	Total		
TrustMark	61	2.508	1.621	4.129	1.000 / 1.000 / 1.000	39.660
SSL	64	784.813	1.473	786.286	1.000 / 1.000 / 1.000	39.979
FNNS	64	1623.342	2.351	1625.693	0.990 / 1.000 / 1.000	34.429
DWT-DCT	48	2.725	1.401	4.126	0.960 / 0.990 / 0.770	43.062
Stable Signature	48	20.907	2.166	23.072	1.000 / 0.998 / 0.950	26.219
LaWa	48	26.542	1.499	28.041	1.000 / 1.000 / 1.000	30.798
Ours-DCT	64	**0.857**	**0.441**	**1.298**	1.000 / 1.000 / 1.000	**55.370**

(b) Average Robustness

Post-hoc				Long Payload			
Method	TPR	Bit	Msg	Method	TPR	Bit	Msg
DWT-DCT (48)	4.25%	71.37%	13.60%	MaxMark (8,192)	–	93.63%	0.00%
SSL (64)	87.54%	97.93%	85.64%	**Ours-High (8,192)**	100.00%	**96.65%**	**48.03%**
FNNS (64)	39.93%	90.50%	33.70%	MaxMark (16,384)	–	92.31%	0.00%
TrustMark (61)	96.44%	**99.53%**	**96.03%**	**Ours-High (16,384)**	100.00%	**93.50%**	**22.04%**
Ours-Mid (64)	**99.39%**	98.98%	87.44%	**Ours-High (32,768)**	100.00%	**90.40%**	**4.76%**
Diffusion				**Autoregressive**			
Method	TPR	Bit	Msg	Method	TPR	Bit	Msg
LaWa (48)	**99.95%**	**99.58%**	87.58%	ClusterMark	94.20%	–	–
Stable Signature (48)	94.62%	94.13%	43.53%	WmAR	94.78%	–	–
Ours-Mid (48)	99.73%	99.33%	**92.76%**	**Ours-Mid**	**98.39%**	–	–

Table 4: Efficiency and average robustness comparisons. (a) Watermark embedding, detection, and total runtime on 200 Stable Diffusion v1.5 images, together with the corresponding detection, decoding, and image quality results. All runtimes are reported in seconds. (b) Average robustness over the corresponding image perturbations. Post-hoc, diffusion, and long-payload settings report TPR at an FPR of 1%, while the autoregressive setting reports TPR at an FPR of 0.1%.

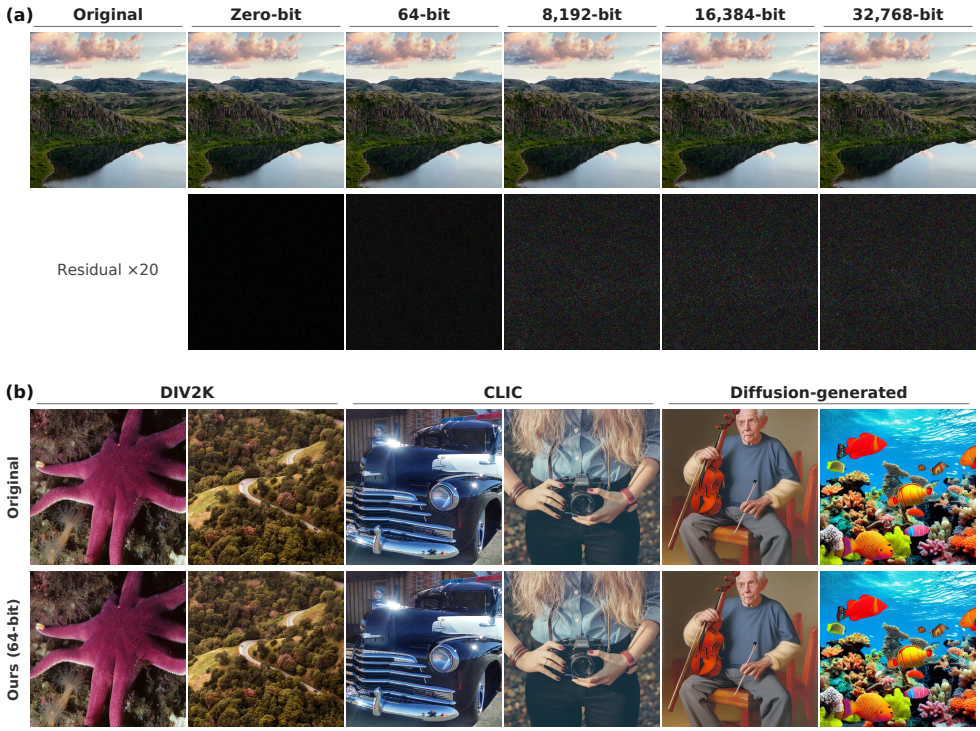

Figure 3: Qualitative results across different watermark settings and image sources. (a) Zero-bit, 64-bit, and high-capacity multi-bit watermarking on the same input image. The zero-bit setting uses the Gaussian construction, the 64-bit setting uses the Binary construction with DCT-Mid, and the 8,192-, 16,384-, and 32,768-bit settings use the Binary construction with DCT-High. Pixel residuals $|x^{\mathrm{wm}} - x|$ are amplified by $20\times$ using the same visualization range. (b) Results on DIV2K, CLIC, and diffusion-generated images using the same 64-bit DCT-Mid setting, with two examples from each image source.

4 Conclusion

We presented UniMark, a unified image watermarking framework for zero-bit detection, multi-bit detection, and multi-bit decoding. By combining statistical watermark constructions with likelihood-ratio tests, UniMark supports reliable detection and high-capacity message decoding while preserving image quality. The framework operates directly on existing images without accessing or modifying the generator. Future work will extend UniMark to a broader range of applications and increase the number of embedded bits while maintaining image fidelity and decoding accuracy.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the analysis and interpretation of some experimental results. Additionally, we used generative AI tools for language editing and paper organization. All AI-assisted content was reviewed by the authors, and experimental claims were verified against the actual experimental results. All reported experimental results were obtained and verified by the authors. We take responsibility for the final content of this work, including text and claims produced or revised with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the code required to reproduce the main experiments in the supplementary material. Detailed experimental settings, hyperparameters, and implementation details are provided in Appendices C and C.3. The configurations of the compared baselines are documented in Appendix C.4.

REFERENCES

Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

N. Ahmed, T. Natarajan, and K.R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974. doi: 10.1109/T-C.1974.223784.

N. Alexandridis and A. Klinger. Real-time walsh-hadamard transformation. *IEEE Transactions on Computers*, 21(3):288–292, 1972. doi: 10.1109/TC.1972.5008951. URL https://doi.org/10.1109/TC.1972.5008951.

Mauro Barni, Franco Bartolini, Vito Cappellini, and Alessandro Piva. A dct-domain system for robust image watermarking. *Signal Processing*, 66(3):357–372, 1998. ISSN 0165-1684. doi: 10.1016/S0165-1684(98)00015-2. URL https://www.sciencedirect.com/science/article/pii/S0165168498000152.

Black Forest Labs. FLUX.1-schnell. https://huggingface.co/black-forest-labs/FLUX.1-schnell, 2024.

Black Forest Labs. FLUX.2: Analyzing and enhancing the latent space of flux – representation comparison, 2025. URL https://bfl.ai/research/representation-comparison.

Tu Bui, Shruti Agarwal, and John Collomosse. TrustMark: Robust watermarking and watermark removal for arbitrary resolution images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18629–18639, 2025.

Xuanhang Chang, Zhonghao Yang, Cheng Zhuo, and Yu Li. Maxmark: High-capacity diffusion-native watermarking via robust and invertible latent embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9394–9403, June 2026.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934. ISSN 0006-3444. doi: 10.1093/biomet/26.4.404. URL https://doi.org/10.1093/biomet/26.4.404.

A. Cohen, Ingrid Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45(5):485–560, 1992. doi: 10.1002/cpa.3160450502. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160450502.

Ronald R. Coifman, Yves Meyer, Steven Quake, and M. Victor Wickerhauser. Signal processing and compression with wavelet packets. In J. S. Byrnes, Jennifer L. Byrnes, Kathryn A. Hargreaves, and Karl Berry (eds.), *Wavelets and Their Applications*, pp. 363–379. Springer Dordrecht, 1994. doi: 10.1007/978-94-011-1028-0_18. URL https://link.springer.com/chapter/10.1007/978-94-011-1028-0_18.

James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1965. doi: 10.1090/S0025-5718-1965-0178586-1.

Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996, 1988. doi: 10.1002/cpa.3160410705. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160410705.

Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. Watermarking images in self-supervised latent spaces. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022. URL https://arxiv.org/abs/2112.09581.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22466–22477, 2023.

Alfred Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910. doi: 10.1007/BF01456326.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, 44(13):800–801, June 2008. ISSN 0013-5194. doi: 10.1049/el:20080522.

Nikola Jovanović, Ismail Labiad, Tomáš Souček, Martin Vechev, and Pierre Fernandez. Watermarking autoregressive image generation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/67cef321196556c4f1e17989ed7a8adb-Abstract-Conference.html.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Changhoon Kim, Kyle Min, Maitreya Patel, Sheng Cheng, and Yezhou Yang. WOUAF: Weight modulation for user attribution and fingerprinting in text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8974–8983, 2024.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. URL https://iclr.cc/archive/2014/conference-proceedings/.

Varsha Kishore, Xiangyu Chen, Yan Wang, Boyi Li, and Kilian Q. Weinberger. Fixed neural network steganography: Train the images, not the network. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=hcMvApxGSzZ.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Nils Lukas and Florian Kerschbaum. PTW: Pivotal tuning watermarking for pre-trained image generators. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 2241–2258. USENIX Association, 2023. URL https://www.usenix.org/conference/usenixsecurity23/presentation/lukas.

Denis Lukovnikov, Andreas Müller, Erwin Quiring, and Asja Fischer. ClusterMark: Towards robust watermarking for autoregressive image generators with visual token clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9213–9222, 2026.

Henrique S. Malvar and Gary J. Sullivan. YCoCg-R: A color space with rgb reversibility and low dynamic range. Technical Report JVT-I014r3, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, 2003. URL https://www.microsoft.com/en-us/research/publication/ycocg-r-a-color-space-with-rgb-reversibility-and-low-dynamic-range/.

Jerzy Neyman and Egon S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231:289–337, 1933. doi: 10.1098/rsta.1933.0009.

Ahmad Rezaei, Mohammad Akbari, Saeed Ranjbar Alvar, Arezou Fatemi, and Yong Zhang. LaWa: Using latent space for in-generation image watermarking. In *European Conference on Computer Vision (ECCV)*, 2024. doi: 10.1007/978-3-031-73024-5_8.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1874–1883, June 2016. doi: 10.1109/CVPR.2016.207. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Shi_Real-Time_Single_Image_CVPR_2016_paper.html.

ShieldMnt. Invisible watermark. https://github.com/ShieldMnt/invisible-watermark, 2021. Software implementation of DWT-DCT image watermarking.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation, 2024. URL https://arxiv.org/abs/2406.06525.

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

George Toderici, Wenzhe Shi, Radu Timofte, Lucas Theis, Johannes Ballé, Eirikur Agustsson, Nick Johnston, and Fabian Mentzer. Workshop and challenge on learned image compression (clic2020), 2020. URL https://www.compression.cc/.

Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Advances in Neural Information Processing Systems*, volume 36, pp. 58047–58063. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/b54d1757c190ba20dbc4f9e4a2f54149-Paper-Conference.pdf.

Shmuel Winograd. On computing the discrete fourier transform. *Mathematics of Computation*, 32 (141):175–199, 1978. ISSN 0025-5718. doi: 10.1090/S0025-5718-1978-0468306-4.

Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian Shading: Provable performance-lossless image watermarking for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12162–12171, 2024.

Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14448–14457, 2021.

Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18431–18441, October 2025.

Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. HiDDeN: Hiding data with deep networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 657–672, 2018.

APPENDIX CONTENTS

A Related Work **15**

A.1 Post-hoc Image Watermarking 15

A.2 Generator-Integrated Image Watermarking 15

A.3 Statistical Watermark Detection and Decoding 15

B Theoretical Analysis **16**

B.1 Gaussian Construction 16

B.2 Optimality under Independence 16

B.2.1 Binary Construction 17

B.2.2 Gaussian Construction 18

B.3 Optimality Gap under Dependence 20

B.3.1 Detection Power Gap 20

B.3.2 Decoding Optimality Gap 22

C Additional Experimental Details **23**

C.1 Evaluation Metrics and Protocols 23

C.2 Calibration and Detection Threshold 24

C.3 UniMark Implementation Details 24

C.4 Baseline Configurations 25

D Additional Results **26**

D.1 Generalization across Representations 26

D.2 Further Analysis of High-Capacity Watermarking 27

D.3 Visual Comparison of Watermark Residuals 27

D.4 Robustness 28

D.5 Additional Results of Zero-Bit Watermarking 30

A Related Work

Table 5 summarizes representative image watermarking methods in terms of insertion stage, deployment requirements, payload, and image quality. Existing methods differ substantially in their training requirements, dependence on the generator, and supported payload capacity. In contrast, UniMark requires neither watermark-specific training nor generator access, while supporting both zero-bit and high-capacity multi-bit watermarking within the same framework.

A.1 Post-hoc Image Watermarking

Post-hoc watermarking embeds watermarks directly into existing images without accessing their generation process. Traditional methods typically modify image coefficients in predefined transform domains, such as DCT-based watermarking (Barni et al., 1998) and DWT-DCT (ShieldMnt, 2021). Recent methods further use feature representations or neural networks to optimize or learn the watermark embedding process. SSL (Fernandez et al., 2022) optimizes images in a pretrained self-supervised feature space, while FNNS (Kishore et al., 2022) keeps the decoder fixed and directly optimizes images to encode messages. TrustMark (Bui et al., 2025) trains a watermark embedder and extractor for message embedding and recovery. These post-hoc approaches rely on predefined transforms, feature representations, per-image optimization, or trained watermarking networks. Our method instead calibrates projection statistics from different reconstructable representations into a shared quantile space and applies the same watermark construction in this space. This allows the framework to use a unified embedding and verification procedure across different reconstructable representations without training a watermarking network.

A.2 Generator-Integrated Image Watermarking

Generator-integrated watermarking embeds watermarks by modifying components or procedures within the image generation pipeline. For diffusion models, Stable Signature (Fernandez et al., 2023) and LaWa (Rezaei et al., 2024) embed watermarks through the decoder, while Gaussian Shading (Yang et al., 2024) and MaxMark (Chang et al., 2026) embed watermark information in the latent space. For GAN generators, PTW (Lukas & Kerschbaum, 2023) embeds watermarks by adapting the generator parameters. For autoregressive image generation, WmAR (Jovanović et al., 2025) and ClusterMark (Lukovnikov et al., 2026) introduce watermark signals during visual-token sampling. These methods can exploit the internal structure of a specific generator, but their embedding and verification procedures are typically tied to the corresponding generation pipeline. Applying the same watermarking approach across different generation paradigms therefore often requires designs specific to the corresponding model or generation process. Our method operates on reconstructable representations after image generation and does not require access to generator parameters, latent sampling, or model-specific generation components. This allows the same watermarking framework to be applied to real images and images generated by diffusion, autoregressive, and GAN models.

A.3 Statistical Watermark Detection and Decoding

Statistical watermarking methods construct detectable signals by controlling the distributions of selected random variables. Tree-Ring (Wen et al., 2023) embeds structured signals into the initial noise of diffusion models and detects them from the inverted noise. Gaussian Shading (Yang et al., 2024) encodes watermarks into diffusion latents while preserving their standard Gaussian marginal distribution, enabling watermark construction from a known latent distribution. WmAR (Jovanović et al., 2025) extends statistical watermarking to autoregressive image generation by modifying visual-token sampling distributions. Their statistical constructions rely on random variables provided by specific generation processes, such as diffusion noise, diffusion latents, or autoregressive tokens. Our framework instead starts from projection statistics in any reconstructable representation and calibrates each statistic into a shared quantile space using unwatermarked data. The Binary and Gaussian constructions then introduce controlled distribution shifts in this space and yield explicit likelihood ratios for zero-bit detection, multi-bit detection, and bit-wise decoding. The same statistical watermark construction can therefore be applied across different image representations and generation sources without relying on a specific generative distribution.

Method	Insertion Stage	Deployment			Capacity & Quality	
		Train-free	Gen.-free	Gen.-agnostic	Max Payload	PSNR @ Max
DWT-DCT (ShieldMnt, 2021)	RGB post-hoc	✓	✓	✓	48	38.86
SSL (Fernandez et al., 2022)	RGB post-hoc	×	✓	✓	256	33.07
FNNS (Kishore et al., 2022)	RGB post-hoc	✓	✓	✓	1,024	29.82
TrustMark (Bui et al., 2025)	RGB post-hoc	×	✓	✓	61	39.78
Gaussian Shading (Yang et al., 2024)	Diffusion noise	✓	×	×	256	–
LaWa (Rezaei et al., 2024)	Diffusion decoder	×	×	×	48	31.04
Stable Signature (Fernandez et al., 2023)	Diffusion decoder	×	×	×	48	25.95
MaxMark (Chang et al., 2026)	Diffusion latent	×	×	×	16,384	–
PTW (Lukas & Kerschbaum, 2023)	GAN weights	×	×	×	100	–
WmAR (Jovanović et al., 2025)	AR sampling	✓	×	×	0	–
ClusterMark (Lukovnikov et al., 2026)	AR sampling	×	×	×	0	–
Ours	**RGB post-hoc**	✓	✓	✓	**32,768**	**42.18**

Table 5: Comparison of watermarking methods by insertion stage, deployment requirements, payload, and image quality. Insertion Stage indicates where the watermark is embedded; Train-free indicates no watermark-specific training; Gen.-free indicates no generator access during embedding; and Gen.-agnostic indicates independence from a specific generator. Max Payload is the largest evaluated effective message payload; PSNR @ Max reports the PSNR at that payload.

B Theoretical Analysis

B.1 Gaussian Construction

The calibrated quantile q_k is mapped through the inverse cumulative distribution function of the standard Gaussian distribution to define the continuous statistic $Y_k = \Phi^{-1}(q_k)$, where Φ denotes the standard Gaussian cumulative distribution function. Since $q_k \sim \mathcal{U}(0,1)$ under H_0, we have $Y_k \mid H_0 \sim \mathcal{N}(0,1)$.

A strength parameter $\beta > 0$ controls the magnitude of the shift applied to the Gaussian statistic. A positive shift sets $Y_k^{\mathrm{wm}} = Y_k + \beta$ and gives $q_k^{\mathrm{wm}} = \Phi(\Phi^{-1}(q_k) + \beta)$, while a negative shift sets $Y_k^{\mathrm{wm}} = Y_k - \beta$ and gives $q_k^{\mathrm{wm}} = \Phi(\Phi^{-1}(q_k) - \beta)$.

For zero-bit watermarking, a fixed positive shift is applied to all statistics, giving $Y_k \sim \mathcal{N}(\beta, 1)$ under H_1 and $Y_k \sim \mathcal{N}(0,1)$ under H_0. From these marginal distributions, the independence-form log-likelihood ratio is

$$\log \Lambda_{\mathcal{G}}^{\mathrm{zero}}(\mathbf{Y}) = \beta \sum_{k=1}^{K} Y_k - \frac{K\beta^2}{2}. \tag{8}$$

For multi-bit watermarking, we partition the K statistic indices into B disjoint sets $\{\mathcal{I}_b\}_{b=1}^{B}$, each containing M indices and assigned to one message bit m_b. For $k \in \mathcal{I}_b$, a positive shift is applied when $m_b = 0$ and a negative shift when $m_b = 1$, giving $Y_k \mid m_b = 0 \sim \mathcal{N}(\beta, 1)$ and $Y_k \mid m_b = 1 \sim \mathcal{N}(-\beta, 1)$, respectively. Under independence, the log-likelihood ratio between $m_b = 1$ and $m_b = 0$ for the b-th message bit is

$$\log \Lambda_{\mathcal{G}_b}^{\mathrm{multi,dec}} = \sum_{k \in \mathcal{I}_b} \log \frac{p(Y_k \mid m_b = 1)}{p(Y_k \mid m_b = 0)} = -2\beta R_b, \qquad R_b = \sum_{k \in \mathcal{I}_b} Y_k. \tag{9}$$

Thus, $\hat{m}_b = 0$ when $R_b \geq 0$, and $\hat{m}_b = 1$ when $R_b < 0$. Detailed likelihood derivations for multi-bit detection and decoding, together with their optimality proofs, are provided in Appendix B.2.2.

B.2 Optimality under Independence

Under the ideal independence assumption, the joint distributions for zero-bit detection and bit-wise decoding factorize as

$$p(\mathbf{Y} \mid H_h) = \prod_{k=1}^{K} p(Y_k \mid H_h), \qquad p(\mathbf{Y}_b \mid m_b) = \prod_{k \in \mathcal{I}_b} p(Y_k \mid m_b), \tag{10}$$

where $h \in \{0,1\}$ and $\mathbf{Y}_b = \{Y_k : k \in \mathcal{I}_b\}$ denotes the statistics assigned to the b-th message bit.

B.2.1 BINARY CONSTRUCTION

Zero-bit Watermark Detection. In the Binary construction, the statistic is defined as $Y_k = \mathbb{I}(q_k > 1/2)$. Under H_0, $q_k \sim \mathcal{U}(0,1)$, and hence $Y_k \sim \text{Bernoulli}(1/2)$. For zero-bit watermarking, we select the right half as the target region, giving $P(Y_k = 1 \mid H_1) = p_\beta$ and $P(Y_k = 0 \mid H_1) = \rho_\beta$. Thus, the distributions of a single statistic under the two hypotheses are

$$Y_k \mid H_0 \sim \text{Bernoulli}\left(\frac{1}{2}\right), \qquad Y_k \mid H_1 \sim \text{Bernoulli}(p_\beta). \tag{11}$$

Under the independence assumption, the joint likelihood factorizes into the product of the marginal likelihoods. Let $C = \sum_{k=1}^{K} Y_k$ denote the number of quantiles falling in the right half. The joint likelihoods are

$$p(\mathbf{Y} \mid H_0) = \left(\frac{1}{2}\right)^K, \qquad p(\mathbf{Y} \mid H_1) = p_\beta^C \rho_\beta^{K-C}. \tag{12}$$

The likelihood ratio of H_1 against H_0 is therefore

$$\Lambda\mathcal{B}^{\text{zero}}(\mathbf{Y}) = \frac{p(\mathbf{Y} \mid H_1)}{p(\mathbf{Y} \mid H_0)} = 2^K p_\beta^C \rho_\beta^{K-C}. \tag{13}$$

Taking the logarithm gives

$$\log \Lambda\mathcal{B}^{\text{zero}}(\mathbf{Y}) = C \log \frac{p_\beta}{\rho_\beta} + K \log(2\rho_\beta). \tag{14}$$

Since $p_\beta/\rho_\beta = e^\beta$ and $\beta > 0$, the log-likelihood ratio is strictly increasing in C. Accordingly, C follows $\text{Binomial}(K, 1/2)$ under H_0 and $\text{Binomial}(K, p_\beta)$ under H_1. By the Neyman–Pearson lemma, for any fixed false-positive rate, an upper-tail test on C is equivalent to thresholding the full likelihood ratio and achieves the maximum detection power. Therefore, the Binary zero-bit detection rule is Neyman–Pearson optimal under the independence assumption.

Multi-bit Watermark Detection. For multi-bit watermark detection, H_0 denotes the unwatermarked case, while H_1 denotes the multi-bit watermarked case with an unknown message. We assume that the message bits are independent and that $m_b = 0$ and $m_b = 1$ are equally likely. For the statistics assigned to the b-th message bit, let $C_b = \sum_{k \in \mathcal{I}_b} Y_k$ denote the number of quantiles satisfying $q_k > 1/2$. Under H_0, each Y_k follows $\text{Bernoulli}(1/2)$, giving

$$p(\mathbf{Y}_b \mid H_0) = 2^{-M}. \tag{15}$$

Under H_1, the unknown message bit is marginalized out. Using the conditional likelihoods under $m_b = 0$ and $m_b = 1$, we obtain

$$\begin{aligned} p(\mathbf{Y}_b \mid H_1) &= \frac{1}{2} p(\mathbf{Y}_b \mid m_b = 0) + \frac{1}{2} p(\mathbf{Y}_b \mid m_b = 1) \\ &= \frac{1}{2} p_\beta^{C_b} \rho_\beta^{M-C_b} + \frac{1}{2} \rho_\beta^{C_b} p_\beta^{M-C_b}. \end{aligned} \tag{16}$$

The likelihood ratio for the statistics assigned to the b-th message bit is therefore

$$\Lambda\mathcal{B}_b^{\text{multi,det}} = 2^{M-1} \left(p_\beta^{C_b} \rho_\beta^{M-C_b} + \rho_\beta^{C_b} p_\beta^{M-C_b} \right). \tag{17}$$

Using $p_\beta/\rho_\beta = e^\beta$, this likelihood ratio can be written as

$$\Lambda\mathcal{B}_b^{\text{multi,det}} = \frac{\cosh(\beta(C_b - M/2))}{\cosh^M(\beta/2)}. \tag{18}$$

Under independence across the statistics assigned to different message bits, the log-likelihood ratio for the full multi-bit watermark is the sum of the bit-wise log-likelihood ratios:

$$\log \Lambda\mathcal{B}^{\text{multi,det}}(\mathbf{Y}) = \sum_{b=1}^{B} \left[\log \cosh(\beta(C_b - M/2)) - M \log \cosh(\beta/2) \right]. \tag{19}$$

Since $\cosh(\cdot)$ is symmetric around zero and increases with its absolute argument, the statistic measures how far each C_b deviates from $M/2$. By the Neyman–Pearson lemma, thresholding this likelihood ratio achieves the maximum detection power at any fixed false-positive rate.

Multi-bit Watermark Decoding. For multi-bit watermarking, the b-th message bit m_b is carried by the M statistics indexed by $\mathcal{I}_b$. When $m_b = 0$, the right half is the target region, giving $P(Y_k = 1 \mid m_b = 0) = p_\beta$ and $P(Y_k = 0 \mid m_b = 0) = \rho_\beta$. When $m_b = 1$, the left half is the target region, giving $P(Y_k = 1 \mid m_b = 1) = \rho_\beta$ and $P(Y_k = 0 \mid m_b = 1) = p_\beta$.

Let $C_b = \sum_{k \in \mathcal{I}_b} Y_k$ denote the number of quantiles satisfying $q_k > 1/2$ among the indices $k \in \mathcal{I}_b$. Under the independence assumption, the joint likelihoods conditioned on $m_b = 0$ and $m_b = 1$ are

$$p(\mathbf{Y}_b \mid m_b = 0) = p_\beta^{C_b} \rho_\beta^{M - C_b}, \qquad p(\mathbf{Y}_b \mid m_b = 1) = \rho_\beta^{C_b} p_\beta^{M - C_b}, \tag{20}$$

where $\mathbf{Y}_b = \{Y_k : k \in \mathcal{I}_b\}$.

The log-likelihood ratio between $m_b = 1$ and $m_b = 0$ for the b-th message bit is

$$\begin{aligned} \log \Lambda\mathcal{B}_b^{\text{multi,dec}} &= \log \frac{p(\mathbf{Y}_b \mid m_b = 1)}{p(\mathbf{Y}_b \mid m_b = 0)} \\ &= C_b \log \frac{\rho_\beta}{p_\beta} + (M - C_b) \log \frac{p_\beta}{\rho_\beta} = \beta(M - 2C_b), \end{aligned} \tag{21}$$

where we use $p_\beta / \rho_\beta = e^\beta$.

Maximum-likelihood decoding selects the bit value with the larger conditional likelihood, giving $\hat{m}_b = 1$ when $\log \Lambda\mathcal{B}_b^{\text{multi,dec}} > 0$ and $\hat{m}_b = 0$ when $\log \Lambda\mathcal{B}_b^{\text{multi,dec}} \leq 0$. Equivalently, the bit is decoded as $\hat{m}_b = 1$ when $C_b < M/2$, and as $\hat{m}_b = 0$ when $C_b \geq M/2$.

Since different message bits are assigned disjoint index sets and the corresponding statistics are conditionally independent under the independence assumption, the likelihood of the full message factorizes across message bits:

$$p(\mathbf{Y} \mid m) = \prod_{b=1}^{B} p(\mathbf{Y}_b \mid m_b). \tag{22}$$

Therefore, the maximum-likelihood estimate of the full message decomposes into the bit-wise maximum-likelihood decisions:

$$\hat{m}_{\text{ML}} = \arg \max_{m \in \{0,1\}^B} p(\mathbf{Y} \mid m) = (\hat{m}_1, \ldots, \hat{m}_B)\,. \tag{23}$$

B.2.2 Gaussian Construction

Zero-bit Watermark Detection. In the Gaussian construction, each statistic Y_k follows a standard Gaussian distribution under the unwatermarked case, while zero-bit watermarking applies a positive shift of β to each statistic. Thus, the distributions of Y_k under the two hypotheses are

$$Y_k \mid H_0 \sim \mathcal{N}(0, 1), \qquad Y_k \mid H_1 \sim \mathcal{N}(\beta, 1). \tag{24}$$

Under the independence assumption, the joint likelihood factorizes into the product of the marginal likelihoods. The joint likelihoods under the two hypotheses are

$$\begin{aligned} p(\mathbf{Y} \mid H_0) &= (2\pi)^{-K/2} \exp\left(-\frac{1}{2} \sum_{k=1}^{K} Y_k^2\right), \\ p(\mathbf{Y} \mid H_1) &= (2\pi)^{-K/2} \exp\left(-\frac{1}{2} \sum_{k=1}^{K} (Y_k - \beta)^2\right). \end{aligned} \tag{25}$$

The likelihood ratio of H_1 against H_0 is

$$\begin{aligned}\Lambda\mathcal{G}^{\text{zero}}(\mathbf{Y}) &= \frac{p(\mathbf{Y} \mid H_1)}{p(\mathbf{Y} \mid H_0)} \\ &= \exp\left[-\frac{1}{2}\sum_{k=1}^{K}\left((Y_k-\beta)^2 - Y_k^2\right)\right] \\ &= \exp\left[-\frac{1}{2}\sum_{k=1}^{K}\left(-2\beta Y_k + \beta^2\right)\right] \\ &= \exp\left(\beta\sum_{k=1}^{K} Y_k - \frac{K\beta^2}{2}\right).\end{aligned} \tag{26}$$

Taking the logarithm and defining $R = \sum_{k=1}^{K} Y_k$ gives

$$\log \Lambda\mathcal{G}^{\text{zero}}(\mathbf{Y}) = \beta R - \frac{K\beta^2}{2}. \tag{27}$$

Since $\beta > 0$, the log-likelihood ratio is strictly increasing in R. Accordingly, $R \sim \mathcal{N}(0, K)$ under H_0 and $R \sim \mathcal{N}(K\beta, K)$ under H_1. By the Neyman–Pearson lemma, for any fixed false-positive rate, an upper-tail test on R is equivalent to thresholding the full likelihood ratio and achieves the maximum detection power. Therefore, the Gaussian zero-bit detection rule is Neyman–Pearson optimal under the independence assumption.

Multi-bit Watermark Detection. For multi-bit watermark detection, H_0 denotes the unwatermarked case, while H_1 denotes the multi-bit watermarked case with an unknown message. We assume that the message bits are independent and that $m_b = 0$ and $m_b = 1$ are equally likely. For the statistics assigned to the b-th message bit, let $R_b = \sum_{k\in\mathcal{I}_b} Y_k$. Under H_0, $Y_k \sim \mathcal{N}(0,1)$, and hence $R_b \sim \mathcal{N}(0, M)$. Under H_1, $R_b \sim \mathcal{N}(M\beta, M)$ when $m_b = 0$, and $R_b \sim \mathcal{N}(-M\beta, M)$ when $m_b = 1$. Marginalizing over the unknown message bit gives

$$p(R_b \mid H_1) = \frac{1}{2}\mathcal{N}(R_b; M\beta, M) + \frac{1}{2}\mathcal{N}(R_b; -M\beta, M), \tag{28}$$

while $p(R_b \mid H_0) = \mathcal{N}(R_b; 0, M)$.

The likelihood ratio for the statistics assigned to the b-th message bit is

$$\begin{aligned}\Lambda\mathcal{G}_b^{\text{multi,det}} &= \frac{p(R_b \mid H_1)}{p(R_b \mid H_0)} \\ &= \frac{1}{2}\exp\left(\beta R_b - \frac{M\beta^2}{2}\right) + \frac{1}{2}\exp\left(-\beta R_b - \frac{M\beta^2}{2}\right) \\ &= \exp\left(-\frac{M\beta^2}{2}\right)\cosh(\beta R_b).\end{aligned} \tag{29}$$

Taking the logarithm gives

$$\log \Lambda\mathcal{G}_b^{\text{multi,det}} = \log\cosh(\beta R_b) - \frac{M\beta^2}{2}. \tag{30}$$

Under independence across the statistics assigned to different message bits, the log-likelihood ratio for the full multi-bit watermark is

$$\log \Lambda\mathcal{G}^{\text{multi,det}}(\mathbf{Y}) = \sum_{b=1}^{B}\left[\log\cosh(\beta R_b) - \frac{M\beta^2}{2}\right]. \tag{31}$$

Since $\log\cosh(\beta R_b)$ increases with $|R_b|$, the statistic measures the deviation of each R_b from zero. By the Neyman–Pearson lemma, thresholding this likelihood ratio achieves the maximum detection power at any fixed false-positive rate.

Multi-bit Watermark Decoding. For the b-th message bit, a positive shift is applied to the corresponding statistics when $m_b = 0$, and a negative shift when $m_b = 1$. Thus, for $k \in \mathcal{I}_b$, the conditional distributions under the two bit values are

$$Y_k \mid m_b = 0 \sim \mathcal{N}(\beta, 1), \qquad Y_k \mid m_b = 1 \sim \mathcal{N}(-\beta, 1). \tag{32}$$

Let $R_b = \sum_{k \in \mathcal{I}_b} Y_k$ denote the sum of the statistics indexed by $\mathcal{I}_b$. Under the independence assumption, the joint likelihoods conditioned on $m_b = 0$ and $m_b = 1$ are

$$\begin{aligned} p(\mathbf{Y}_b \mid m_b = 0) &= (2\pi)^{-M/2} \exp\left[-\frac{1}{2}\sum_{k \in \mathcal{I}_b} (Y_k - \beta)^2\right], \\ p(\mathbf{Y}_b \mid m_b = 1) &= (2\pi)^{-M/2} \exp\left[-\frac{1}{2}\sum_{k \in \mathcal{I}_b} (Y_k + \beta)^2\right]. \end{aligned} \tag{33}$$

The log-likelihood ratio between $m_b = 1$ and $m_b = 0$ for the b-th message bit is

$$\begin{aligned} \log \Lambda\mathcal{G}_b^{\text{multi,dec}} &= \log \frac{p(\mathbf{Y}_b \mid m_b = 1)}{p(\mathbf{Y}_b \mid m_b = 0)} \\ &= -\frac{1}{2}\sum_{k \in \mathcal{I}_b} \left[(Y_k + \beta)^2 - (Y_k - \beta)^2\right] \\ &= -2\beta \sum_{k \in \mathcal{I}_b} Y_k = -2\beta R_b. \end{aligned} \tag{34}$$

Maximum-likelihood decoding selects the bit value with the larger conditional likelihood, giving $\hat{m}_b = 1$ when $\log \Lambda\mathcal{G}_b^{\text{multi,dec}} > 0$ and $\hat{m}_b = 0$ when $\log \Lambda\mathcal{G}_b^{\text{multi,dec}} \leq 0$. Since $\beta > 0$, this is equivalent to decoding $\hat{m}_b = 1$ when $R_b < 0$ and $\hat{m}_b = 0$ when $R_b \geq 0$.

Since different message bits are assigned disjoint index sets and the corresponding statistics are conditionally independent under the independence assumption, the likelihood of the full message factorizes across message bits:

$$p(\mathbf{Y} \mid m) = \prod_{b=1}^{B} p(\mathbf{Y}_b \mid m_b). \tag{35}$$

Therefore, the maximum-likelihood estimate of the full message is obtained from the bit-wise maximum-likelihood decisions:

$$\hat{m}_{\text{ML}} = \arg \max_{m \in \{0,1\}^B} p(\mathbf{Y} \mid m) = (\hat{m}_1, \ldots, \hat{m}_B). \tag{36}$$

B.3 Optimality Gap under Dependence

B.3.1 Detection Power Gap

When dependence exists among the statistics, the true joint likelihood ratio contains the dependence correction $\Delta(\mathbf{Y})$ defined in the main text. Specifically, the full log-likelihood ratio satisfies

$$\log \Lambda^*(\mathbf{Y}) = \log \Lambda_{\text{ind}}(\mathbf{Y}) + \Delta(\mathbf{Y}). \tag{37}$$

Here, Λ^* is based on the true joint distributions and accounts for dependence among the statistics, while the practical detector uses Λ_{ind} obtained by omitting Δ. Thus, Λ^* corresponds to the full likelihood ratio required by the unknown Neyman–Pearson optimal test, whereas Λ_{ind} is the statistic used by the practical detector.

For a target false-positive rate α, we compare the optimal and practical detectors under the same FPR constraint. Let ϕ_α^* denote the Neyman–Pearson optimal detector among all detectors with false-positive rate at most α, and let ϕ denote the practical detector based on Λ_{ind}. The comparison is made under

$$P_{H_0}(\phi_\alpha^*(\mathbf{Y}) = 1) \leq \alpha, \qquad P_{H_0}(\phi(\mathbf{Y}) = 1) = \text{FPR}_\phi \leq \alpha. \tag{38}$$

Construction	Payload	H_0 Audit		H_1 Evaluation		
		FP / 20,480	U_{FP}	FN / 2,048	Power ↑	Gap Bound ↓
Binary	Zero-bit	1	2.72×10^{-4}	0	1.000000	1.80×10^{-3}
	32	5	5.70×10^{-4}	0	1.000000	1.80×10^{-3}
	64	5	5.70×10^{-4}	0	1.000000	1.80×10^{-3}
	128	6	6.38×10^{-4}	0	1.000000	1.80×10^{-3}
	512	2	3.53×10^{-4}	0	1.000000	1.80×10^{-3}
	1024	5	5.70×10^{-4}	0	1.000000	1.80×10^{-3}
Gaussian	Zero-bit	2	3.53×10^{-4}	0	1.000000	1.80×10^{-3}
	32	8	7.70×10^{-4}	0	1.000000	1.80×10^{-3}
	64	2	3.53×10^{-4}	0	1.000000	1.80×10^{-3}
	128	5	5.70×10^{-4}	0	1.000000	1.80×10^{-3}
	512	10	8.98×10^{-4}	0	1.000000	1.80×10^{-3}
	1024	11	9.61×10^{-4}	3	0.998535	4.275×10^{-3}

Table 6: Finite-sample guarantees for watermark presence detection at the target false-positive rate $\alpha = 10^{-3}$. U_{FP} and U_{FN} denote the one-sided upper confidence bounds on the false-positive and false-negative rates, respectively. All configurations satisfy $U_{\mathrm{FP}} \leq \alpha$, and the resulting power-gap guarantee is $G_\alpha \leq U_{\mathrm{FN}}$.

Their detection powers are defined under H_1 as

$$\pi^*_\alpha = P_{H_1}(\phi^*_\alpha(\mathbf{Y}) = 1)\,, \qquad \pi_\phi = P_{H_1}(\phi(\mathbf{Y}) = 1)\,. \tag{39}$$

We define the power gap between the practical detector and the Neyman–Pearson optimal detector as $G_\alpha = \pi^*_\alpha - \pi_\phi$. When $\mathrm{FPR}_\phi \leq \alpha$, the Neyman–Pearson lemma gives $\pi_\phi \leq \pi^*_\alpha$, and hence $G_\alpha \geq 0$. Since the detection power of any detector cannot exceed 1, we have $\pi^*_\alpha \leq 1$. Therefore,

$$0 \leq G_\alpha = \pi^*_\alpha - \pi_\phi \leq 1 - \pi_\phi = \mathrm{FNR}_\phi\,. \tag{40}$$

This bound allows the gap to the optimal detection power to be controlled without explicitly estimating the true joint distribution or the Neyman–Pearson optimal detector. The condition $\mathrm{FPR}_\phi \leq \alpha$ ensures that the practical detector satisfies the same false-positive rate constraint, while FNR_ϕ directly upper-bounds the power gap. With finite samples, these quantities can be further controlled through confidence upper bounds on the false-positive and false-negative rates.

Finite-Sample Guarantee. We evaluate zero-bit detection and multi-bit presence detection with $B \in \{32, 64, 128, 512, 1024\}$ under both the Binary and Gaussian constructions, resulting in 12 detection configurations. For each configuration, the detection threshold is fixed before the final evaluation with a target false-positive rate of $\alpha = 10^{-3}$. We use 20,480 H_0 samples to evaluate the false-positive rate and 2,048 H_1 samples to evaluate the detection power and false-negative rate.

Let U_{FP} and U_{FN} denote the one-sided Clopper–Pearson upper confidence bounds (Clopper & Pearson, 1934) on the true false-positive and false-negative rates, respectively. We use 97.5% one-sided confidence bounds for both FPR and FNR. By the Bonferroni correction, the two bounds jointly hold with at least 95% confidence for each configuration. Therefore, when $U_{\mathrm{FP}} \leq \alpha$, the practical detector satisfies the target false-positive rate constraint at the corresponding confidence level.

From Eq. 40, the power gap between the practical detector and the Neyman–Pearson optimal detector is upper-bounded by the true false-negative rate. When $U_{\mathrm{FP}} \leq \alpha$, the practical detector satisfies the target false-positive rate constraint, while the finite-sample confidence bound gives $\mathrm{FNR}_\phi \leq U_{\mathrm{FN}}$. Therefore, the finite-sample power-gap guarantee is

$$0 \leq G_\alpha \leq \mathrm{FNR}_\phi \leq U_{\mathrm{FN}}. \tag{41}$$

This bound does not require explicit estimation of the true joint distribution or the unknown Neyman–Pearson optimal detector.

As shown in Table 6, all 12 detection configurations satisfy $U_{\mathrm{FP}} \leq 10^{-3}$ at the target false-positive rate $\alpha = 10^{-3}$. Among them, 11 configurations produce no false negatives among the 2,048 H_1

Construction	Payload	All-Bit Evaluation	Finite-Sample Guarantee
		BitAcc ↑	G_{bit} Upper Bound ↓
Binary	32	100.0000%	0.2475%
	64	100.0000%	0.2475%
	128	99.9986%	0.2475%
	512	99.8475%	0.5145%
	1024	99.2750%	1.3309%
Gaussian	32	100.0000%	0.2475%
	64	100.0000%	0.2475%
	128	99.9997%	0.2475%
	512	99.8154%	0.6182%
	1024	98.9659%	1.5552%

Table 7: Finite-sample guarantees for bit decoding across different payload sizes. BitAcc is computed over all decoded bits, while the finite-sample guarantee is obtained by pre-selecting one random bit from each image. Since $G_{\text{bit}} \leq \text{BER} \leq U_{\text{BER}}$, the last column reports the corresponding upper bound on the bit-level optimality gap.

samples, giving $U_{\text{FN}} = 1.80 \times 10^{-3}$ and $G_\alpha \leq 1.80 \times 10^{-3}$. For Gaussian multi-bit detection with $B = 1024$, three false negatives are observed, yielding an empirical detection power of 0.998535 and $U_{\text{FN}} = 4.275 \times 10^{-3}$. Its power gap to the unknown Neyman–Pearson optimal detector is therefore bounded by $G_\alpha \leq 4.275 \times 10^{-3}$.

B.3.2 Decoding Optimality Gap

When the statistics are dependent, the practical decoder based on the bit-wise likelihood ratio under the independence assumption may differ from the optimal bit decoder under the true joint distribution. Under a uniform bit prior, the optimal decoder selects each bit according to its true joint likelihood. Let a_{bit} denote the average bit accuracy of the practical decoder, and let a^*_{bit} denote that of the corresponding optimal decoder.

We define the bit-level decoding optimality gap as

$$G_{\text{bit}} = a^*_{\text{bit}} - a_{\text{bit}}. \tag{42}$$

Since the optimal bit accuracy is no smaller than the practical bit accuracy and cannot exceed 1, the gap is directly bounded by the practical bit error rate:

$$0 \leq G_{\text{bit}} \leq 1 - a_{\text{bit}} = \text{BER}\,. \tag{43}$$

Therefore, a small BER implies a small bit-level optimality gap. This bound does not require estimating the true joint distribution or explicitly computing the optimal bit decoder.

Finite-Sample Guarantee. For both the Binary and Gaussian constructions, we evaluate multi-bit decoding with $B \in \{32, 64, 128, 512, 1024\}$. Each configuration includes four image sources: Stable Diffusion v1.5, LlamaGen, StyleGAN2, and FLUX.1-schnell, with 2,048 images from each source and 8,192 test images in total. To avoid treating correlated bits from the same image as independent samples, we pre-select one bit position uniformly at random for each image. Each image therefore provides one bit-error indicator for the finite-sample analysis.

Let p_s denote the true bit error rate for source s, and let U_s denote its one-sided Clopper–Pearson upper confidence bound. We use a 99.375% one-sided confidence bound for each source. By the Bonferroni correction, the four source-wise bounds hold jointly with at least 97.5% confidence. With equal weights over the four sources, the overall BER and its upper bound satisfy

$$\text{BER} = \frac{1}{4}\sum_{s=1}^{4} p_s \leq \frac{1}{4}\sum_{s=1}^{4} U_s \equiv U_{\text{BER}}. \tag{44}$$

Combining Eq. 43 with the finite-sample bound above gives

$$0 \leq G_{\text{bit}} \leq \text{BER} \leq U_{\text{BER}}. \tag{45}$$

Thus, U_{BER} directly gives a finite-sample upper bound on the gap between the practical and optimal bit decoders. Since each image contributes only one randomly selected bit, the finite-sample analysis does not treat multiple bit errors from the same image as independent observations.

As shown in Table 7, for $B \leq 128$, both the Binary and Gaussian constructions maintain nearly perfect bit accuracy, with finite-sample optimality gaps bounded by 0.2475%. As the payload increases to $B = 512$ and $B = 1024$, the bit accuracy decreases slightly, with larger corresponding optimality-gap bounds. At the highest capacity of $B = 1024$, the Binary and Gaussian constructions achieve bit accuracies of 99.2750% and 98.9659%, with gap bounds of 1.3309% and 1.5552%, respectively.

C Additional Experimental Details

C.1 Evaluation Metrics and Protocols

For watermark presence detection, we report TPR at a target FPR of 0.1% for standard evaluation and 1% for robustness evaluation, except for the autoregressive robustness comparison, which uses an FPR of 0.1%. For multi-bit watermarking, we use BitAcc for bit-wise decoding accuracy and MessageAcc for full-message recovery accuracy. For image quality, we use peak signal-to-noise ratio (PSNR) (Huynh-Thu & Ghanbari, 2008) and structural similarity index measure (SSIM) (Wang et al., 2004) to measure watermark-induced distortion, and use $\Delta\mathrm{FID}$ (Heusel et al., 2017) when paired image comparisons are not applicable. For detectors requiring empirical thresholds, the thresholds are calibrated on independent unwatermarked samples and fixed during evaluation; details of threshold calibration and baseline detection protocols are provided in Appendices C.2 and C.4, respectively.

FID Evaluation. We define $\Delta\mathrm{FID} = \mathrm{FID}_{\mathrm{wm}} - \mathrm{FID}_{\mathrm{clean}}$, where both FID values are computed against the same real-image reference distribution. For Table 2(b), FID is computed using $50{,}000$ generated images per method and the ImageNet-256 reference statistics used in the official RAR evaluation (Yu et al., 2025). For Table 3, images are generated from the same $1{,}000$ COCO captions, with the corresponding $1{,}000$ real COCO images used as the common reference set.

Image Quality Comparison Protocol. For Figure 2(a), for each method and supported payload, we select the tested configuration with the highest mean PSNR among those satisfying the predefined performance requirements. We require $\mathrm{TPR} \geq 99\%$ for zero-bit watermarking and $\mathrm{BitAcc} \geq 99\%$ with $\mathrm{MessageAcc} > 90\%$ for multi-bit watermarking. All selected operating points are evaluated on the same $1{,}000$ images.

Efficiency Measurement. Table 4(a) reports runtime over 200 Stable Diffusion v1.5 images at 512×512 resolution with a batch size of 1. GPU-based methods run on an NVIDIA A800 80GB GPU, while DWT-DCT runs on an AMD EPYC 7543 CPU. Timing includes watermark embedding, detection, and message decoding, and excludes model initialization, warm-up, file I/O, and metric computation. GPU synchronization is performed before and after each timed stage. For Stable Signature and LaWa, embedding time includes only watermarked VAE decoding from preloaded final latents and excludes diffusion sampling.

Figure 1 Evaluation Details. Figure 1(a) compares image quality across different image sources. The first seven axes correspond to DIV2K, CLIC, MS-COCO, Stable Diffusion v1.5, FLUX.1-schnell, LlamaGen, and StyleGAN2, and report the mean paired RGB PSNR between original and watermarked images. These axes range from 0 to 60 dB, with larger values indicating higher image quality. The eighth axis reports FID for RAR-XL/256 against ImageNet-256 using a separate reversed scale, with 38 at the outer boundary and 44 at the center, so points farther from the center indicate lower FID. FID is computed using $1{,}000$ images for each method. Because PSNR and FID use different metrics and scales, the enclosed radar area does not represent a unified performance measure. For the first seven axes, we show only multi-bit configurations satisfying $\mathrm{TPR} \geq 99\%$, $\mathrm{BitAcc} \geq 99\%$, and $\mathrm{MessageAcc} > 90\%$. On the FID axis, multi-bit methods are required to satisfy $\mathrm{BitAcc} \geq 99\%$ and $\mathrm{MessageAcc} > 90\%$, while zero-bit methods are required to satisfy $\mathrm{TPR} \geq 99\%$ at an FPR of 0.1%. Results that do not satisfy the corresponding requirements are

omitted rather than assigned a value of zero. Numbers in parentheses in the legend indicate the watermark payload in bits.

Figure 1(b) compares full-message recovery accuracy across payload sizes on 512×512 Stable Diffusion v1.5 images. The horizontal axis uses equally spaced discrete payloads of 61, 128, 256, 512, 1,024, 2,048, 4,096, and 8,192 bits. Except for MaxMark, line styles are determined by the mean PSNR of the corresponding evaluated payload. A segment is solid when both endpoints have mean PSNR ≥ 40 dB and dashed when either endpoint has mean PSNR < 40 dB. This division uses 40 dB only as the boundary and imposes no additional lower PSNR bound on the < 40 dB group. MaxMark is shown using its reported MessageAcc without a PSNR constraint. Marker size represents the total watermark injection and extraction time over 200 images for each method, using logarithmic scaling with runtime. The same marker size is used for all payloads of a method, so the runtime does not represent separate timing at each payload.

C.2 Calibration and Detection Threshold

Our framework uses two calibration stages with different purposes, both based only on unwatermarked H_0 images. In the first stage, we estimate the empirical distribution G_k^0 of each projection score S_k under H_0 and map it to the shared quantile space as $q_k = G_k^0(S_k)$. This marginal calibration aligns the one-dimensional distributions across directions but does not require independence among the resulting statistics.

In the second stage, we determine the threshold of the final watermark-presence detection statistic using another independent set of H_0 images. Even when each q_k is approximately uniform under H_0, residual dependence may remain among the statistics, so the H_0 distribution of the final detection statistic can differ from its ideal form under independence. Therefore, for a target FPR α, we determine the threshold directly from the empirical upper-tail distribution of the final detection scores on independent H_0 samples. The threshold is selected so that no more than $\lfloor \alpha n_{\mathrm{thr}} \rfloor$ calibration samples are classified as watermarked, where n_{thr} is the number of H_0 samples used for threshold calibration. This empirical calibration accounts for residual dependence among the statistics and finite-sample errors in the CDF estimation when setting the operating threshold.

For the main experiments, the estimation of G_k^0 and the detection-threshold calibration each use 16,384 independent unwatermarked images. Each set contains 4,096 images from each of Stable Diffusion v1.5, LlamaGen-XL, StyleGAN2, and FLUX.1-schnell. The two calibration sets are disjoint from each other and from the final test set. At the target FPR of 0.1%, at most 16 of the 16,384 threshold-calibration samples are allowed to exceed the threshold, giving an empirical calibration FPR no greater than 0.1%. The robustness experiments follow the same threshold-calibration procedure using their corresponding target FPR. A separate threshold is calibrated for each detector configuration and representation and remains fixed during evaluation.

Detection-threshold calibration is used only for watermark-presence detection, while multi-bit message decoding directly uses the likelihood-ratio decision boundaries derived in Appendix B.2. The Binary construction uses $C_b = M/2$ as the bit-wise decision boundary, while the Gaussian construction uses $R_b = 0$. After fixing the detection thresholds, we further use an independent set of 20,480 unwatermarked images as the H_0 audit set for the finite-sample FPR guarantee. As shown in Table 6, all detector configurations satisfy $U_{\mathrm{FP}} \leq \alpha = 10^{-3}$ on this independent audit set.

C.3 UniMark Implementation Details

DCT Representation. We first normalize RGB pixel values to $[0, 1]$ and then apply an orthonormal two-dimensional DCT separately to each color channel. For a 512×512 image, we define the normalized frequency radius as

$$r(u, v) = \frac{\sqrt{u^2 + v^2}}{511\sqrt{2}}, \qquad u, v \in \{0, \ldots, 511\}. \tag{46}$$

Ours-Mid uses coefficients satisfying $1/3 < r(u, v) \leq 2/3$, while Ours-High uses those satisfying $r(u, v) > 2/3$. The key-dependent orthonormal projection directions are restricted to the selected frequency subspace.

Table 8: Watermark configurations used in the zero-bit and multi-bit experiments. B denotes the payload length, M the number of projection directions per bit, K the total number of projection directions, and β the watermark strength. For multi-bit configurations, $K = BM$; for zero-bit configurations, $B = 0$ and M is denoted by "-". Values of β are rounded to four decimal places.

Setting	Variant	Construction	B	M	K	β
Zero-bit	Ours-High	Gaussian	0	-	4,096	0.8500
	Ours-High	Binary	0	-	8,192	2.2500
Post-hoc	Ours-Mid	Binary	64	6,419	410,816	0.3526
	Ours-High	Binary	64	3,715	237,760	0.7439
Diffusion	Ours-Mid	Binary	48	8,559	410,832	0.3526
	Ours-High	Binary	48	4,953	237,744	0.7439
High-capacity	Ours-High	Binary	8,192	29	237,568	4.0759
	Ours-High	Binary	16,384	14	229,376	4.3962
	Ours-High	Binary	32,768	7	229,376	4.3962

Watermark Configurations. Table 8 summarizes the watermark configurations used in the zero-bit, post-hoc, diffusion, and high-capacity experiments. For a B-bit multi-bit watermark, each bit is assigned M projection directions, giving $K = BM$ directions in total; zero-bit watermarking directly uses K projection directions. In the cross-source comparison, each configuration is kept fixed across all image sources. For the 48-bit diffusion comparison, we adjust M to keep the total number of directions K approximately unchanged from the corresponding 64-bit configuration, while retaining the same strength parameter β. The high-capacity configurations are selected using a separate 300-image parameter-selection experiment and are reused without retuning on the Stable Diffusion v1.5 test images. The zero-bit experiments use the DCT-High representation, with the Gaussian construction used in the main experiments and the Binary construction further evaluated in the additional experiments.

Key-dependent Orthonormal Directions. The secret key κ determines a permutation of the DCT coefficient positions within the selected frequency subspace and is used to deterministically generate the orthonormal directions. The permuted coordinates are divided into disjoint groups of size 256. For each group, we generate a standard Gaussian random matrix and apply QR decomposition to obtain an orthogonal matrix, whose columns are used as the directions within that group. We normalize the column signs according to the diagonal of the R factor, ensuring that the same key reproduces the same directions. Directions within each group are orthonormal by construction, while directions from different groups have disjoint supports and are therefore orthogonal. For a fixed key, the embedder and detector can reconstruct and cache the same coordinate permutation and orthonormal directions for repeated use.

Storage and Computation. All directions are deterministically generated from the secret key κ through the coordinate permutation and orthogonal blocks, so the full set of directions does not need to be explicitly stored. Given the same key, the embedder and detector can independently reconstruct the same directions, which can also be cached after initialization for efficient repeated use. The runtime in Table 4(a) is measured after the directions and calibration data have been initialized, and therefore reports the computation required for online watermark embedding and detection.

C.4 Baseline Configurations

Post-hoc Baselines. DWT-DCT (ShieldMnt, 2021) uses the `dwtDct` implementation from the invisible-watermark library, with 48 message bits and 16 fixed detection bits; a watermark is detected when at least 15 detection bits are recovered correctly. SSL (Fernandez et al., 2022) uses the released DINO ResNet-50 feature extractor and embeds 64-bit messages through 40 image optimization steps; detection uses the mean log probability of the predicted message bits. FNNS (Kishore et al., 2022) uses a fixed randomly initialized decoder with adaptive average pooling to 8×8 for 64 bit predictions, and uses the same mean log-probability detection score. TrustMark (Bui et al., 2025) uses the released Q model with a watermark strength of 1.0 and BCH-5 coding for 61 message bits;

its detection score measures agreement between the decoder output and the BCH codeword obtained by re-encoding the recovered message. Detection for SSL, FNNS, and TrustMark does not use the ground-truth message, and their thresholds are calibrated on independent unwatermarked samples and fixed during evaluation.

Diffusion Baselines. The comparison with Stable Signature (Fernandez et al., 2023) and LaWa (Rezaei et al., 2024) in Table 2(a) uses Stable Diffusion v1.5. Stable Signature uses a separately fine-tuned decoder for each registered message, while LaWa uses its released checkpoint. Each method is evaluated with 10 registered 48-bit messages on images corresponding to the same $1{,}000$ sampled latents, yielding $10{,}000$ watermarked images. For a common detection protocol, a watermark is declared present when the decoded message matches any registered message in at least 38 bit positions. Under the assumption that decoded bits are independent and equally likely under H_0, this rule bounds the overall false-positive rate across the 10 registered messages below 0.1%.

Autoregressive Baselines. WmAR (Jovanović et al., 2025) and ClusterMark (Lukovnikov et al., 2026) are evaluated using their official implementations on RAR-XL at a resolution of 256×256. Detection is evaluated on $1{,}000$ generated images per method, with a positive decision when the corresponding detector returns a p-value no greater than 10^{-3}.

High-Capacity Baseline. MaxMark (Chang et al., 2026) uses the released $8{,}192$-bit and $16{,}384$-bit checkpoints for Stable Diffusion v1.5. Message extraction recovers the latent through 50-step DDIM (Song et al., 2021) inversion using an empty prompt and a guidance scale of 1.0. No error-correcting codes or message repetition are used, and BitAcc and MessageAcc are computed directly from the original messages.

D Additional Results

D.1 Generalization across Representations

To evaluate whether the framework generalizes across different reconstructable representations, we instantiate the transform T with a range of conventional transforms and learned representations. These representations include the discrete cosine transform (DCT) (Ahmed et al., 1974), block discrete cosine transform (Block DCT) (Ahmed et al., 1974), fast Fourier transform (FFT) (Cooley & Tukey, 1965), Haar discrete wavelet transform (Haar DWT) (Haar, 1910), Daubechies discrete wavelet transform (Daubechies DWT) (Daubechies, 1988), Cohen–Daubechies–Feauveau 5/3 discrete wavelet transform (CDF 5/3 DWT) (Cohen et al., 1992), wavelet packet transform (Coifman et al., 1994), Walsh–Hadamard transform (Alexandridis & Klinger, 1972), Pixel Unshuffle (Shi et al., 2016), reversible YCoCg color transform (YCoCg-R) (Malvar & Sullivan, 2003), and the variational autoencoder (VAE)(Kingma & Welling, 2014) latent representation of FLUX.2 (Black Forest Labs, 2025). For each representation, we keep the statistical projection, H_0 calibration, and Binary or Gaussian construction unchanged, and only re-estimate the corresponding H_0 distributions, detection thresholds, and strength parameter β. For zero-bit watermarking, we evaluate both the Binary and Gaussian constructions on 500 images generated by LlamaGen-XL; for multi-bit watermarking, we evaluate 64-bit Binary watermarks on 300 DIV2K images. All zero-bit detectors target an FPR of 0.1%, with detection thresholds recalibrated using independent H_0 samples for each representation.

As shown in Table 9, the framework maintains reliable zero-bit detection across a wide range of reconstructable representations. Under the Gaussian construction, all representations achieve 100% TPR; under the Binary construction, all reach 100% except FLUX.2 VAE, which achieves 99.80%. Image quality varies across representations, with frequency- and wavelet-domain representations generally achieving high PSNR and SSIM while maintaining reliable detection. In particular, the db4/HH subspace of Daubechies DWT reaches PSNR values of 69.45 dB and 64.88 dB under the Gaussian and Binary constructions, respectively.

As shown in Table 10, for each representation, we scan the watermark strength in the clean setting and preferentially report an operating point satisfying $\mathrm{PSNR} \geq 45$ dB and at least 99% TPR, BitAcc, and MessageAcc. For representations satisfying these constraints, the detection and decoding metrics are therefore typically close to saturation, and the main differences lie in the image

Representation	Subspace	Gaussian			Binary		
		TPR@0.1% ↑	PSNR ↑	SSIM ↑	TPR@0.1% ↑	PSNR ↑	SSIM ↑
FLUX.2 VAE	Full latent	100.00%	30.00	0.9365	99.80%	34.69	0.9648
DCT	Mid frequency	100.00%	54.25	0.9979	100.00%	58.22	0.9992
Block DCT	8×8 / High	100.00%	62.34	0.9997	100.00%	64.09	0.9998
FFT	High frequency	100.00%	55.72	0.9985	100.00%	59.98	0.9992
Haar DWT	HH	100.00%	67.93	0.9999	100.00%	63.23	0.9998
Daubechies DWT	db4 / HH	100.00%	69.45	0.9999	100.00%	64.88	0.9998
CDF 5/3 DWT	HH	100.00%	65.67	0.9999	100.00%	62.93	0.9997
Wavelet Packet	High	100.00%	66.18	0.9999	100.00%	64.53	0.9998
Walsh–Hadamard	High sequency	100.00%	59.87	0.9994	100.00%	61.29	0.9996
Pixel Unshuffle	All phases	100.00%	50.78	0.9961	100.00%	50.72	0.9959
YCoCg-R + DCT	Cg	100.00%	57.32	0.9988	100.00%	56.86	0.9989

Table 9: Zero-bit watermarking across different reconstructable representations. A representative subspace is reported for each representation. Both Gaussian and Binary detectors use independently calibrated thresholds at an FPR of 0.1%.

Representation	Subspace	Detection and Decoding			Image Quality	
		TPR@0.1% ↑	BitAcc ↑	MessageAcc ↑	PSNR ↑	SSIM ↑
FLUX.2 VAE	Full latent	100.00%	99.99%	99.67%	29.30	0.9137
DCT	High frequency	100.00%	99.99%	99.67%	50.16	0.9965
Block DCT	8×8 / High	100.00%	99.98%	99.67%	50.29	0.9966
FFT	High frequency	100.00%	99.98%	99.33%	52.59	0.9980
Haar DWT	HH	100.00%	99.95%	99.00%	52.69	0.9980
Daubechies DWT	db2 / HH	100.00%	99.98%	99.33%	52.91	0.9981
CDF 5/3 DWT	HH	100.00%	99.99%	99.33%	51.29	0.9974
Wavelet Packet	High	100.00%	99.94%	99.00%	52.78	0.9981
Walsh–Hadamard	High sequency	100.00%	99.99%	99.33%	52.72	0.9980
Pixel Unshuffle†	All phases	100.00%	98.88%	61.67%	45.02	0.9885
YCoCg-R + DCT†	Cg	100.00%	99.92%	97.67%	45.01	0.9875

Table 10: Generalization of 64-bit multi-bit watermarking across different reconstructable representations. A representative subspace is reported for each representation, using the Binary construction throughout. For each representation, we scan the watermark strength and preferentially report a clean operating point satisfying $\mathrm{PSNR} \geq 45$ dB and at least 99% TPR, BitAcc, and MessageAcc. If no operating point satisfies all constraints within the scanned range, the best available point is reported and marked with †.

quality required to achieve this performance. Overall, the results show that the same Binary statistical construction can be applied across a range of reconstructable representations, while the choice of representation and subspace affects the trade-off among detection, decoding, and image quality.

D.2 Further Analysis of High-Capacity Watermarking

To further analyze the variation in complete-message recovery under long payloads, we compare the decoding results of different images in the $32{,}768$-bit setting. The results show clear differences in high-capacity decoding difficulty across images. Images with fewer high-frequency details generally support longer error-free messages, while images containing dense textures, fine lines, text, or strong-contrast edges are more likely to produce decoding errors.

As shown in Figure 4, images with fewer high-frequency structures achieve error-free decoding in the $32{,}768$-bit setting, whereas images with more complex high-frequency structures exhibit more bit errors. This trend is consistent with watermark embedding in the high-frequency DCT subspace, indicating that the high-frequency complexity of the image affects decoding performance under long payloads.

D.3 Visual Comparison of Watermark Residuals

Figure 5 compares the pixel residuals introduced by different watermarking methods on the same input image, with all methods achieving successful watermark detection and full message recovery.

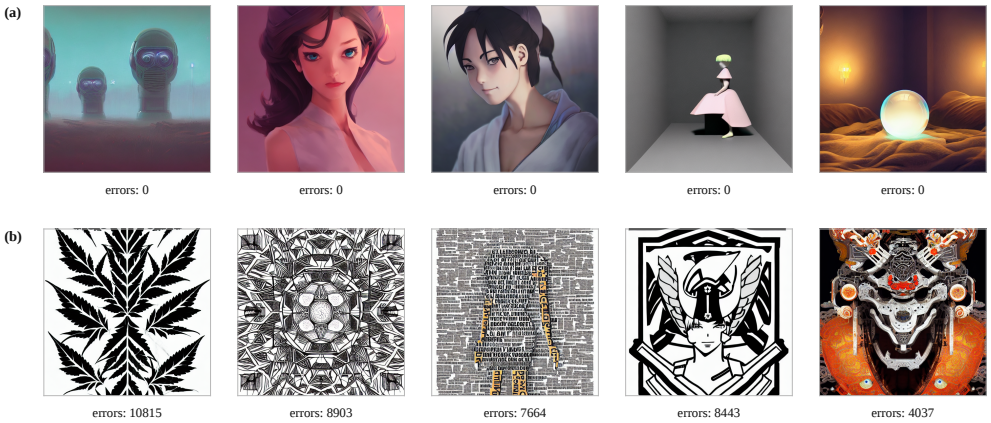

Figure 4: Representative examples of 32,768-bit watermark decoding under different levels of high-frequency image complexity. Panel (a) shows images with fewer high-frequency details, all of which achieve complete message recovery, while panel (b) shows images with more complex high-frequency structures and different numbers of decoding errors.

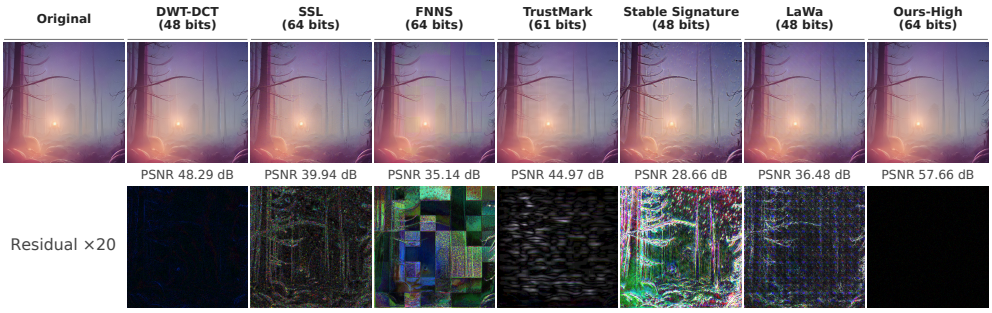

Figure 5: Visual comparison of watermark residuals across different methods on the same input image. The first row shows the watermarked images and their corresponding PSNR values, while the second row shows the pixel residuals $|x^{\mathrm{wm}} - x|$ amplified by $20\times$ using the same visualization range. All shown methods successfully detect the watermark and fully recover the embedded message.

Under the same $20\times$ amplification and visualization range, several baselines exhibit pronounced texture, block, or edge structures in their residuals. Ours-High yields substantially weaker residuals and achieves the highest PSNR of 57.66 dB. Under matched detection and decoding performance, these results indicate lower image distortion for Ours-High.

D.4 Robustness

To comprehensively evaluate watermark robustness under common image processing operations, we consider JPEG compression, resizing, Gaussian blur, brightness changes, Gaussian noise, contrast changes, and combined attacks. The evaluated settings include JPEG Q80; resizing factors of 1.00, 0.95, 0.89, 0.84, 0.77, 0.71, 0.63, and 0.55, where the image is first resized by the corresponding factor and then restored to its original resolution; Gaussian blur with $(k = 3, \sigma = 0.5)$ and $(k = 7, \sigma = 1)$; brightness ranges of $U[0.75, 1.25]$, $U[0.5, 1.5]$, and $U[0.5, 2]$; Gaussian noise with $\sigma = 0.05$ and $\sigma = 0.1$; and contrast factors of 0.8, 0.5, and 0.2. We further consider two sequential combined attacks: Comb-1 applies Resize 0.84 followed by Contrast 0.5, while Comb-2 applies Resize 0.84, Gaussian blur with $(k = 5, \sigma = 1.0)$, and Contrast 0.5 in sequence, with the resized image restored to its original resolution before the subsequent operations. All detection thresholds are calibrated in advance using independent clean H_0 samples and remain fixed across all perturbations, without attack-specific threshold adjustment during evaluation. For post-hoc, diffusion, and high-capacity watermarking experiments, the detection threshold is calibrated at an FPR of

Attack	DWT-DCT-IW (48)			SSL-DINO (64)			FNNS-R-adapted (64)			TrustMark-Q (61)			Ours-Mid (64)		
	TPR (%)	Bit (%)	Msg (%)	TPR (%)	Bit (%)	Msg (%)	TPR (%)	Bit (%)	Msg (%)	TPR (%)	Bit (%)	Msg (%)	TPR (%)	Bit (%)	Msg (%)
JPEG Q80	2.51	51.83	0.00	84.87	99.21	77.55	4.55	87.37	1.63	99.73	99.98	99.73	87.40	84.54	1.56
Resize 1.00	2.66	85.98	25.51	99.56	99.94	98.30	99.82	100.00	99.88	99.95	100.00	99.95	100.00	100.00	99.91
Resize 0.95	2.66	86.81	27.72	99.34	99.93	98.39	85.64	99.51	81.57	99.95	100.00	99.95	100.00	99.95	97.74
Resize 0.89	2.63	86.93	27.83	99.04	99.92	98.16	65.51	98.60	58.01	99.95	100.00	99.95	100.00	99.95	97.38
Resize 0.84	2.55	87.07	28.05	99.04	99.92	98.27	48.73	97.62	40.97	99.95	100.00	99.95	100.00	99.93	97.22
Resize 0.77	2.29	87.18	28.25	98.89	99.90	97.93	35.29	96.00	22.92	99.95	100.00	99.95	100.00	99.88	94.88
Resize 0.71	0.71	86.82	25.85	98.77	99.89	97.72	27.29	94.56	15.22	99.95	100.00	99.95	100.00	99.75	90.48
Resize 0.63	0.17	81.29	10.21	98.73	99.88	97.35	20.09	92.78	8.37	99.95	100.00	99.95	99.99	99.32	79.70
Resize 0.55	1.47	71.46	2.33	98.30	99.84	96.50	16.30	90.87	4.79	99.94	100.00	99.94	99.95	97.63	54.74
Gaussian Blur ($k = 3, \sigma = 0.5$)	2.90	87.16	28.69	99.43	99.93	98.40	83.13	99.37	78.79	99.95	100.00	99.95	100.00	99.99	99.38
Gaussian Blur ($k = 7, \sigma = 1$)	2.09	60.98	0.21	98.57	99.84	96.83	11.70	87.91	2.19	99.95	100.00	99.95	100.00	99.55	84.82
Brightness $U[0.75, 1.25]$	2.81	84.85	26.16	97.69	99.78	96.38	95.15	99.43	89.70	99.08	99.90	99.05	100.00	99.97	99.32
Brightness $U[0.5, 1.5]$	2.21	74.19	15.34	94.37	99.46	92.35	89.83	98.61	80.60	96.84	99.51	96.48	99.99	99.96	98.80
Brightness $U[0.5, 2]$	1.96	66.91	11.51	78.20	97.86	73.60	70.32	95.74	58.54	86.62	97.74	85.55	99.94	99.90	98.04
Gaussian Noise ($\sigma = 0.05$)	0.00	76.98	4.68	11.35	92.43	11.79	2.37	81.97	6.47	99.29	99.94	99.29	100.00	99.97	98.28
Gaussian Noise ($\sigma = 0.1$)	0.00	45.17	0.00	0.03	70.10	0.01	0.07	65.17	0.16	79.96	98.12	79.06	100.00	99.64	82.11
Contrast 0.8	3.14	81.49	23.28	99.19	99.91	97.68	64.36	97.62	54.32	99.95	100.00	99.95	100.00	100.00	99.83
Contrast 0.5	7.05	50.22	0.00	97.90	99.87	96.78	6.94	84.68	2.67	99.19	99.94	99.19	100.00	100.00	99.87
Contrast 0.2	36.08	45.66	0.00	92.46	99.49	87.72	1.86	74.30	0.09	66.92	95.26	60.82	100.00	99.99	99.69
Comb-1	6.76	49.93	0.00	97.64	99.82	95.80	4.78	80.69	0.62	99.10	99.92	99.05	100.00	99.93	96.83
Comb-2	6.63	49.80	0.00	94.91	99.64	90.97	4.76	77.79	0.29	99.15	99.91	99.07	99.99	98.76	65.63
Average	**4.25**	**71.37**	**13.60**	**87.54**	**97.93**	**85.64**	**39.93**	**90.50**	**33.70**	**96.44**	**99.53**	**96.03**	**99.39**	**98.98**	**87.44**

Table 11: Detailed robustness comparison with post-hoc watermarking methods. TPR is evaluated using a fixed detection threshold calibrated at an FPR of 1% on clean H_0 samples, while Bit and Msg denote BitAcc and MessageAcc, respectively.

Attack	LaWa (48)			Stable Signature (48)			Ours-Mid (48)		
	TPR (%)	Bit (%)	Msg (%)	TPR (%)	Bit (%)	Msg (%)	TPR (%)	Bit (%)	Msg (%)
JPEG Q80	100.00	99.99	99.50	99.40	92.06	4.23	94.34	88.01	5.30
Resize 1.00	100.00	100.00	100.00	99.98	99.26	81.05	100.00	100.00	100.00
Resize 0.95	100.00	99.99	99.41	99.92	99.01	75.24	100.00	100.00	99.78
Resize 0.89	100.00	99.97	98.43	99.91	98.85	71.79	100.00	100.00	99.85
Resize 0.84	100.00	99.88	94.07	99.87	98.66	67.08	100.00	99.99	99.67
Resize 0.77	100.00	99.67	84.34	99.83	98.23	57.12	100.00	99.98	99.05
Resize 0.71	100.00	99.57	79.55	99.78	97.60	45.12	100.00	99.95	97.94
Resize 0.63	100.00	99.47	74.37	99.63	96.05	24.97	100.00	99.83	94.18
Resize 0.55	100.00	99.36	69.61	99.19	93.24	9.95	100.00	98.92	74.15
Gaussian Blur ($k = 3, \sigma = 0.5$)	100.00	100.00	100.00	99.90	98.90	73.39	100.00	100.00	99.95
Gaussian Blur ($k = 7, \sigma = 1$)	100.00	99.14	58.84	98.23	90.37	2.92	100.00	99.88	95.84
Brightness $U[0.75, 1.25]$	100.00	100.00	100.00	99.95	99.17	78.79	100.00	100.00	99.94
Brightness $U[0.5, 1.5]$	100.00	100.00	100.00	99.89	98.90	73.41	100.00	100.00	99.90
Brightness $U[0.5, 2]$	100.00	100.00	99.99	99.89	98.48	65.42	100.00	99.99	99.55
Gaussian Noise ($\sigma = 0.05$)	100.00	99.63	88.77	98.25	94.66	22.90	100.00	100.00	99.84
Gaussian Noise ($\sigma = 0.1$)	99.38	96.01	47.01	45.21	72.03	0.00	100.00	99.93	96.82
Contrast 0.8	100.00	100.00	100.00	99.92	98.99	74.89	100.00	100.00	100.00
Contrast 0.5	100.00	100.00	100.00	99.70	97.53	50.17	100.00	100.00	100.00
Contrast 0.2	99.75	99.84	98.44	74.60	80.00	0.43	100.00	100.00	100.00
Comb-1	100.00	99.79	90.04	99.12	95.84	35.17	100.00	99.99	99.70
Comb-2	99.89	98.85	56.88	74.80	78.92	0.01	100.00	99.57	86.46
Average	**99.95**	**99.58**	**87.58**	**94.62**	**94.13**	**43.53**	**99.73**	**99.33**	**92.76**

Table 12: Detailed robustness comparison with diffusion generator-integrated watermarking methods. TPR is evaluated using a fixed detection threshold calibrated at an FPR of 1% on clean H_0 samples, while Bit and Msg denote BitAcc and MessageAcc, respectively.

1% on clean H_0 samples, while the autoregressive comparison follows the corresponding baseline evaluation protocol and uses an FPR of 0.1%.

Tables 11, 12, and 13 report the per-attack robustness results under different watermarking settings. In the post-hoc setting, Ours-Mid achieves an average TPR of 99.39%, with TPR remaining above 99.94% under all evaluated perturbations except JPEG Q80. In the diffusion setting, Ours-Mid achieves average TPR, BitAcc, and MessageAcc values of 99.73%, 99.33%, and 92.76%, respectively, while maintaining 100% TPR under all evaluated perturbations except JPEG Q80. In the autoregressive setting, Ours-Mid achieves an average TPR of 98.39%, compared with 94.20% for ClusterMark and 94.78% for WmAR; under the strong Contrast 0.2 perturbation, Ours-Mid still retains a TPR of 99.60%. Overall, our method maintains stable watermark detection under resizing, blur, brightness changes, noise, contrast changes, and combined attacks, while JPEG compression causes a more noticeable degradation in detection and message recovery in some settings.

Table 14 further reports the detection and decoding robustness of high-capacity watermarking under different perturbations. For payloads of $8{,}192$, $16{,}384$, and $32{,}768$ bits, Ours-High maintains 100% TPR across all evaluated conditions, showing stable watermark-presence detection even at very large payloads. At $8{,}192$ and $16{,}384$ bits, Ours-High achieves average BitAcc values of 96.65% and 93.50%, compared with 93.63% and 92.31% for MaxMark. As the payload increases, BitAcc

ATTACK	CLUSTERMARK TPR (%)	WMAR TPR (%)	OURS-MID TPR (%)
JPEG Q80	99.90	99.90	73.60
Resize 1.00	100.00	100.00	100.00
Resize 0.95	100.00	100.00	100.00
Resize 0.89	100.00	100.00	100.00
Resize 0.84	100.00	100.00	100.00
Resize 0.77	99.90	100.00	100.00
Resize 0.71	99.90	100.00	100.00
Resize 0.63	99.90	100.00	100.00
Gaussian Blur ($k = 3, \sigma = 0.5$)	100.00	100.00	100.00
Brightness $U[0.75, 1.25]$	99.50	99.90	100.00
Brightness $U[0.5, 1.5]$	99.10	99.90	100.00
Brightness $U[0.5, 2]$	90.60	98.10	99.70
Gaussian Noise ($\sigma = 0.05$)	88.00	98.20	100.00
Contrast 0.8	100.00	100.00	100.00
Contrast 0.5	98.90	99.80	100.00
Contrast 0.2	27.40	15.70	99.60
Comb-1	98.30	99.80	99.80
Average	**94.20**	**94.78**	**98.39**

Table 13: Detailed robustness comparison with autoregressive generator-integrated watermarking methods. TPR is evaluated using a fixed detection threshold calibrated at an FPR of 0.1% on clean H_0 samples.

ATTACK	MAXMARK (8,192)			OURS-HIGH (8,192)			MAXMARK (16,384)			OURS-HIGH (16,384)			OURS-HIGH (32,768)		
	TPR (%)	Bit (%)	Msg (%)	TPR (%)	Bit (%)	Msg (%)	TPR (%)	Bit (%)	Msg (%)	TPR (%)	Bit (%)	Msg (%)	TPR (%)	Bit (%)	Msg (%)
Clean	–	96.29	0.00	100.00	99.91	92.00	–	94.88	0.00	100.00	99.71	67.70	100.00	99.24	25.50
Resize 1.00	–	96.29	0.00	100.00	99.91	92.00	–	94.88	0.00	100.00	99.71	67.70	100.00	99.24	25.50
Resize 0.95	–	95.55	0.00	100.00	98.52	0.00	–	94.01	0.00	100.00	93.84	0.00	100.00	87.85	0.00
Resize 0.89	–	95.60	0.00	100.00	95.53	0.00	–	94.03	0.00	100.00	88.21	0.00	100.00	81.39	0.00
Resize 0.84	–	95.49	0.00	100.00	89.24	0.00	–	93.92	0.00	100.00	80.34	0.00	100.00	74.00	0.00
Gaussian Blur ($k = 3, \sigma = 0.5$)	–	96.13	0.00	100.00	99.82	60.20	–	94.70	0.00	100.00	99.04	2.40	100.00	96.94	0.00
Brightness $U[0.75, 1.25]$	–	95.75	0.00	100.00	99.81	79.50	–	94.42	0.00	100.00	99.27	39.80	100.00	98.10	6.40
Brightness $U[0.5, 1.5]$	–	95.01	0.00	100.00	99.71	73.40	–	93.76	0.00	100.00	98.94	29.40	100.00	97.39	2.50
Brightness $U[0.5, 2]$	–	92.86	0.00	100.00	99.37	52.70	–	92.15	0.00	100.00	97.82	19.20	100.00	95.26	2.10
Gaussian Noise ($\sigma = 0.05$)	–	78.30	0.00	100.00	84.32	0.00	–	78.27	0.00	100.00	75.60	0.00	100.00	69.80	0.00
Contrast 0.8	–	95.67	0.00	100.00	99.91	90.70	–	94.26	0.00	100.00	99.68	51.70	100.00	99.05	4.40
Contrast 0.5	–	94.63	0.00	100.00	99.91	88.30	–	93.04	0.00	100.00	99.64	30.60	100.00	98.82	0.30
Contrast 0.2	–	89.59	0.00	100.00	99.88	43.60	–	88.17	0.00	100.00	99.08	0.00	100.00	96.48	0.00
Comb-1	–	93.63	0.00	100.00	87.20	0.00	–	91.92	0.00	100.00	78.12	0.00	100.00	72.01	0.00
Average	–	**93.63**	**0.00**	**100.00**	**96.65**	**48.03**	–	**92.31**	**0.00**	**100.00**	**93.50**	**22.04**	**100.00**	**90.40**	**4.76**

Table 14: Detailed robustness results for high-capacity multi-bit watermarking. TPR is evaluated using a fixed detection threshold calibrated at an FPR of 1% on clean H_0 samples, while Bit and Msg denote BitAcc and MessageAcc, respectively. MaxMark does not provide watermark-presence detection results, and its TPR is therefore unavailable.

and MessageAcc become more sensitive to perturbations, with larger degradation under resizing, Gaussian noise, and combined attacks. These results show that watermark-presence detection can remain stable when part of the payload is corrupted, while complete-message recovery degrades more rapidly as the payload length and accumulated bit errors increase.

D.5 ADDITIONAL RESULTS OF ZERO-BIT WATERMARKING

The zero-bit experiments follow the two-stage calibration protocol described in Appendix C.2, where marginal distribution estimation and detection-threshold calibration use two independent sets of unwatermarked H_0 images. Both calibration sets are disjoint from the $3{,}402$ test images, and the detection thresholds are fixed before evaluation.

We further evaluate both Gaussian and Binary zero-bit constructions on real-image datasets and different generative models to examine detection performance and image quality across image sources. Both constructions use the DCT-High representation. As shown in Table 15, both constructions successfully detect all $3{,}402$ watermarked test images and achieve 100% TPR across all image sources. The Gaussian and Binary constructions achieve overall average PSNRs of 57.272 dB and 60.348 dB, with average SSIM values of 0.999014 and 0.999391, respectively. These results show that both constructions support reliable zero-bit watermark detection across different image sources while maintaining high image quality.

Construction	Dataset / Source	#Images	FN	TPR	FP	FPR	PSNR	SSIM
Gaussian	DIV2K	500	0	1.000000	0	0.000000	54.583	0.998675
	CLIC2020	102	0	1.000000	0	0.000000	56.019	0.998767
	MS-COCO	2,000	0	1.000000	1	0.000500	58.376	0.999248
	Stable Diffusion v1.5	200	0	1.000000	1	0.005000	54.946	0.998586
	FLUX.1-schnell	200	0	1.000000	0	0.000000	55.086	0.998228
	LlamaGen-XL	200	0	1.000000	0	0.000000	57.661	0.998887
	StyleGAN2	200	0	1.000000	0	0.000000	57.717	0.998988
	Overall	**3,402**	**0**	**1.000000**	**2**	**0.000588**	**57.272**	**0.999014**
Binary	DIV2K	500	0	1.000000	0	0.000000	56.402	0.999103
	CLIC2020	102	0	1.000000	0	0.000000	58.460	0.999221
	MS-COCO	2,000	0	1.000000	0	0.000000	61.989	0.999561
	Stable Diffusion v1.5	200	0	1.000000	1	0.005000	56.991	0.999061
	FLUX.1-schnell	200	0	1.000000	1	0.005000	57.346	0.998810
	LlamaGen-XL	200	0	1.000000	1	0.005000	60.831	0.999374
	StyleGAN2	200	0	1.000000	0	0.000000	60.634	0.999432
	Overall	**3,402**	**0**	**1.000000**	**3**	**0.000882**	**60.348**	**0.999391**

Table 15: Zero-bit watermarking results across different image sources for the Gaussian and Binary constructions. FN and FP denote the numbers of false negatives and false positives, respectively.