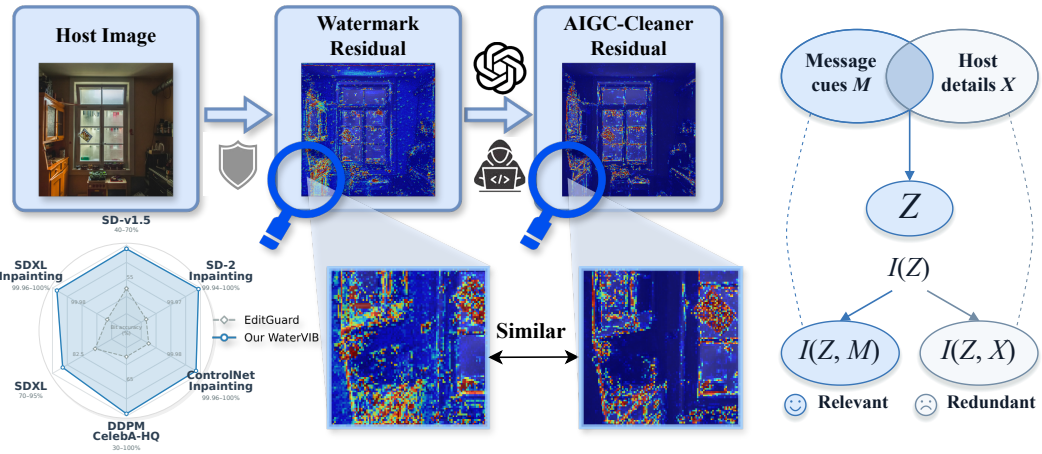
WATERVIB: LEARNING MINIMAL SUFFICIENT WATERMARK REPRESENTATIONS VIA VARIATIONAL INFORMATION BOTTLENECK

Anonymous authors
Paper under double-blind review

Figure 1. From host dependence to message sufficiency. Left: watermark and editing residuals share host structure; the radar plot compares generative-editing robustness. Right: WaterVIB learns a compact representation Z by preserving message information $I(Z, M)$ while reducing image dependence $I(Z, X)$.



ABSTRACT

Robust invisible watermarking is essential for copyright protection, yet generative editing severely threatens watermark survival by rewriting host-image details. Our quantitative analyses reveal that existing watermark representations become deeply entangled with host-specific textures, causing generative modifications to disrupt message decoding. To break this entanglement, we formulate watermark extraction as learning a minimal sufficient statistic for message recovery, thereby substantially reducing the decoder's reliance on fragile host textures. Guided by this insight, we propose WaterVIB, a plug-in variational bottleneck module that decouples message representations from host-image details across existing backbones. Systematic evaluation across five 2D backbones with different architectures under 32 attacks confirms widespread robustness gains thanks to our WaterVIB. On VINE-R, WaterVIB reduces the bit error rate under global purification by 71.0% while increasing embedding PSNR. Furthermore, extending WaterVIB to NeRF-Signature enhances 3D rendering fidelity while preserving clean decoding performance.

1 INTRODUCTION

Robust invisible watermarking embeds recoverable messages into images to support copyright protection and content provenance. Deep watermarking methods jointly train embedding and decoding networks to achieve imperceptible embedding and reliable message recovery (Bui et al., 2023). Training with simulated distortions improves robustness to conventional processing such as Gaussian noise and JPEG compression (Zhu et al., 2018). However, generative image editing remains a challenge (Zhao et al., 2024; Nie et al., 2022). Diffusion-based editing tools use learned generative priors to rewrite image details while maintaining high perceptual quality (Rombach et al., 2022;

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.

Podell et al., 2023). This raises a central question about what information a watermark representation should retain to withstand such editing.

Our quantitative analyses reveal entanglement between watermark representations and host-specific textures. Spectral energy, spatial correlations, and gradient projections show that watermark and editing signals overlap in mid-to-high frequency bands, depend on host content, and exert opposing effects on decoding (Tables 1 and 2). Host textures can conceal watermark signals (Zhang et al., 2020), but editing these textures can disrupt decoding cues. Figure 1 illustrates this vulnerability.

To address this challenge, we seek watermark representations with reduced dependence on host-specific details. Such representations should retain the information needed for message recovery while discarding redundant host information. We formalize this goal as learning a **minimal sufficient statistic (MSS)** of the input for the embedded message. From this formulation, we derive an **information bottleneck** objective that preserves message information while minimizing host dependence (Tishby et al., 2000).

We introduce WaterVIB, which realizes this objective through a variational information bottleneck. WaterVIB is a plug-in training module for existing watermarking networks. We evaluate five 2D watermarking backbones and their WaterVIB counterparts across 32 attacks. These backbones span diverse architectural designs, including convolutional networks, U-Net architectures, invertible networks, and pretrained diffusion-based encoders. Integrating WaterVIB yields notable robustness gains across all five backbones. On VINE-R, WaterVIB reduces BER under global SD1.5 purification by 71.0%, while increasing embedding PSNR from 36.00 to 36.73 dB. On HiDDeN, it reduces BER under DDPM purification by 59.6%. Complementary information measurements and variance analyses provide empirical support for our design principle of preserving message information while reducing host dependence (Figure 4). We further extend WaterVIB to NeRF-Signature, improving 3D rendering fidelity while maintaining clean decoding performance.

Our main contributions are summarized below.

• We propose WaterVIB, a plug-in training module that introduces a variational information bottleneck into existing watermarking networks to improve robustness against generative editing.

• We analyze texture entanglement, derive an information bottleneck objective from minimal sufficient representation learning, and examine message retention and host dependence through a crossed message–host study.

• We improve watermarking performance across five 2D backbones and one 3D backbone through WaterVIB integration. Paired evaluations cover 32 attacks on the 2D backbones and assess rendering fidelity and message recovery on NeRF-Signature.

2 Related Works

2.1 Deep Image Watermarking

Traditional watermarking embeds signals in transform domains to balance imperceptibility and robustness (Langelaar & Lagendijk, 2001; Ganic & Eskicioglu, 2004; Kang et al., 2003). HiDDeN and StegaStamp learn encoder–decoder networks with differentiable noise layers (Zhu et al., 2018; Tancik et al., 2020). Later methods improve fidelity and recovery through invertible networks (Jing et al., 2021) and richer training schemes (Bui et al., 2023). Extensions include lossless host-image recovery with CRMark (Chen et al., 2025), robustness to instruction-driven editing with Robust-Wide (Hu et al., 2024), tamper localization with EditGuard (Zhang et al., 2024), and editing robustness using pretrained generative priors with VINE (Lu et al., 2025). NeRF-Signature further extends watermarking to 3D neural radiance fields (Luo et al., 2025). Despite distortion-aware training, message features can remain dependent on host redundancies (Zhang et al., 2020). We examine how WaterVIB reduces this dependence through message-preserving compression to improve watermarking robustness.

2.2 Information Bottleneck Principle

The Information Bottleneck (IB) principle formulates representation learning as a relevance–compression trade-off (Tishby et al., 2000). Variational IB makes mutual-information bounds

trainable in neural networks through variational inference and reparameterization (Alemi et al., 2016). Alemi et al. (2016) further demonstrate improved generalization and adversarial robustness through this representation-level regularization. Achille & Soatto (2018) theoretically connect the information minimality of sufficient representations to invariance to task-irrelevant nuisance factors. Since then, VIB has been applied to model compression (Razeghi et al., 2022), interpretability (Seo et al., 2023), temporal video analysis (Zhong et al., 2023) and large language model alignment (Yang et al., 2025). Unlike these predominant classification or recognition tasks where the target is a label, we adapt the IB framework to the watermarking channel to construct a signal explicitly decoupled from host image redundancies. WaterVIB uses this connection to derive a message-preserving compression objective for generative watermarking.

3 Theoretical Analysis

To understand the vulnerability described above, we first analyze how watermark signals interact with generative editing. The resulting texture dependence and gradient conflict motivate a representation that retains message information while reducing reliance on the host. We formalize this goal through minimal sufficient statistics and derive an information bottleneck objective.

3.1 Analysis of Vulnerability

Let $\mathcal{X} \subseteq \mathbb{R}^{H\times W\times C}$ and $\mathcal{M}$ be the image and message spaces. A standard encoder $E : \mathcal{X} \times \mathcal{M} \to \mathcal{X}$ produces a watermarked image $x_{wm} = x + \delta$. Unlike traditional additive noise, we model generative purification as a projection onto the natural image space. With $p_{data}(x)$ denoting the natural-image distribution and d a perceptual distance, we write:

$$x_{rec} = \mathcal{G}(x_{wm}) \approx \arg \min_{x' \in \text{supp}(p_{data})} d(x', x_{wm}). \tag{1}$$

Current methods implicitly assume the watermark signal δ resides in a subspace orthogonal to the semantic content. However, we propose that generative models preferentially reconstruct low-frequency semantics while rewriting high-frequency textures where δ resides.

Observation 3.1: Spectral and Spatial Alignment

Let $\mathbf{s}_{atk} = \mathcal{G}(\mathbf{x}_{wm}) - \mathbf{x}_{wm}$ be the purification distortion. We observe that watermark failure is driven by a **Spectral and Spatial Alignment**, where the watermark signal δ and attack distortion s_{atk} overlap in mid-to-high-frequency bands and both exhibit spatial correlations with the host image.

Empirical Validation. Watermark signals in EditGuard (Zhang et al., 2024) and HiDDeN (Zhu et al., 2018) both concentrate over 75% of their energy in the mid-to-high frequency bands (Table 1, left). Natural images typically exhibit approximately $1/f^2$ power spectra (Field, 1987). Compared with the natural-image reference, EditGuard's attack allocates a substantially larger energy fraction to these same bands (43.85% vs. 13.08%). Spatial measurements also reveal a strong correlation between editing perturbations and the host image, with $\rho \approx 0.60$ for EditGuard (Table 1, right).

3.2 Gradient Counter-Optimization

The alignment described in Observation 3.1 leads to a direct interference with the decoding objective. Let $\mathcal{L}(\cdot)$ be the decoding loss, which is the BCE loss between the target message $\mathbf{m}$ and the extracted message (Sander et al., 2025).

Lemma 3.2: Gradient Interference

For a watermarked image x_{wm} optimized to minimize $\mathcal{L}$, the robustness loss under generative attack is approximated by the inner product $\langle s_{atk}, \nabla_x \mathcal{L}(x_{wm})\rangle$. A positive inner product implies *counter-optimization*.

Proof. Consider the watermarked image $\mathbf{x}_{wm} = \mathbf{x}_{host} + \mathbf{s}_{wm}$, and the AIGC purified image $\mathbf{x}_{atk} = \mathbf{x}_{wm} + \mathbf{s}_{atk}$. Assuming $\mathcal{L}$ is differentiable, we apply a first-order Taylor expansion to the

Table 1. Spectral energy distribution and spatial correlation with the host image. Both watermark and editing residuals have larger mid-to-high-frequency energy fractions than the natural-image reference.

Spectral energy distribution (%)				
Signal	**High**	**Mid**	**Low**	**Dominant**
EditGuard				
Watermark δ	30.20	**45.45**	24.35	**Mid–high**
AIGC s_{atk}	23.36	20.49	**56.15**	Low
Natural image x	1.16	11.92	**86.91**	Low
HiDDeN				
Watermark δ	**70.95**	6.08	22.97	**High**
AIGC s_{atk}	**94.86**	1.92	3.22	**High**

Spatial correlation (PCC)		
Signal	**PCC**	**Interpretation**
EditGuard		
AIGC s_{atk}	**.5989**	**Content-dependent**
Watermark δ	.4535	Entangled
Random noise n	$-.0003$	Independent
HiDDeN		
AIGC s_{atk}	$\mathbf{-.3414}$	**Content-dependent**
Watermark δ	.1020	Entangled

loss $\mathcal{L}(\mathbf{x}_{atk})$ around $\mathbf{x}_{wm}$:

$$\mathcal{L}(\mathbf{x}_{atk}) \approx \mathcal{L}(\mathbf{x}_{wm}) + \mathbf{s}_{atk}^{\top} \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}_{wm}) \tag{2}$$

The gradient points toward increasing loss, so a positive attack projection opposes the encoder's decoding objective.

Empirical Validation. Both architectures exhibit the same opposing projection signs (Table 2). Watermark perturbations point toward lower decoding loss, whereas editing perturbations point toward higher loss. For EditGuard, the projection ratio $\gamma = 0.027/0.063 \approx 42.9\%$ quantifies the opposition between watermark and editing perturbations in the first-order loss model.

Table 2. Watermark and AIGC projections onto the decoding-loss gradient.

EditGuard			
Signal	**Projection**	**Direction**	**Effect**
Gradient	—	Target	—
Watermark	$-.063$	Anti-gradient	Minimizes error
AIGC	$\mathbf{+.027}$	Gradient	**Maximizes error**

HiDDeN			
Signal	**Projection**	**Direction**	**Effect**
Gradient	—	Target	—
Watermark	$-.0461$	Anti-gradient	Minimizes error
AIGC	$\mathbf{+.0173}$	Gradient	**Maximizes error**

3.3 Minimal Sufficient Statistic

The preceding analyses identify how editing rewrites host textures that also carry decoding cues. To address this vulnerability, the representation must retain the information needed to recover the message. Recoverability alone, however, still permits dependence on unnecessary host details. We therefore seek a representation that is sufficient for the message and minimal in its dependence on the input. These two requirements lead to the **Minimal Sufficient Statistic (MSS)** of X_{atk} for M.

(1) Message sufficiency. A representation T of X_{atk} must preserve all information relevant to decoding M:

$$I(T; M) = I(X_{atk}; M). \tag{3}$$

(2) Minimality (Disentanglement). To remove information about host textures that is unnecessary for decoding, the representation must also minimize input dependence while preserving message sufficiency:

$$I(T; X_{atk}) \leq I(\widetilde{T}; X_{atk}), \quad \forall \widetilde{T} \text{ s.t. } I(\widetilde{T}; M) = I(X_{atk}; M). \tag{4}$$

The theoretical justification for imposing these constraints is established by the following theorem.

Theorem 3.3: Optimality of MSS Representation

Let the image and message domains be finite and the representation mapping $T = f(X_{atk})$ be deterministic. A representation T that satisfies the Sufficiency (Eq. 3) and the Minimality (Eq. 4) condition is precisely the Minimal Sufficient Statistic (MSS) of X_{atk} for M (A).

Proof. We provide the rigorous derivation in Appendix A.

However, direct computation of the MSS is intractable in high-dimensional spaces. We therefore relax the strict sufficiency constraint to an ϵ-approximate version:

$$\min_{q(t|x_{atk})} I(T; X_{atk}) \qquad \text{s.t.} \quad I(T; M) \geq I(X_{atk}; M) - \epsilon. \tag{5}$$

For an active constraint with a positive Lagrange multiplier λ, the Lagrangian has the Information Bottleneck (IB) form with $\beta = 1/\lambda > 0$:

$$\max_{q(t|x_{atk})} \mathcal{L}_{IB} = I(T; M) - \beta I(T; X_{atk}). \tag{6}$$

Under the extraction chain $(X_{host}, M) \to X_{atk} \to T$, conditional host information satisfies

$$I(T; X_{host} \mid M) \leq I(T; X_{atk}) - I(T; M).$$

Preserving message information while reducing input information therefore tightens the bound on host dependence (Appendix B).

4 Approach

WaterVIB implements T as a stochastic bottleneck output U from backbone features Z. Message prediction and Gaussian regularization jointly optimize U.

4.1 Variational Lower Bounds

We derive differentiable variational bounds for the intractable mutual-information terms in Eq. 6.

Maximizing Relevance ($\mathcal{L}_{rec}$). A variational decoder $r_\psi(\mathbf{m} \mid U)$ gives $I(U; M) \geq H(M) - \mathcal{L}_{rec}$ (Alemi et al., 2016). For bitwise Bernoulli predictions $\hat{\mathbf{m}}$, the reconstruction loss is

$$\mathcal{L}_{rec} = \mathbb{E}[-\log r_\psi(\mathbf{m} \mid U)] = \mathbb{E}[-\mathbf{m}^\top \log \hat{\mathbf{m}} - (\mathbf{1} - \mathbf{m})^\top \log(\mathbf{1} - \hat{\mathbf{m}})]. \tag{7}$$

Minimizing Compression ($\mathcal{L}_{KL}$). Let $q_\alpha(u \mid z)$ denote the stochastic encoder with noise scale α, and $q_\alpha(u) = \mathbb{E}_Z[q_\alpha(u \mid Z)]$ its marginal distribution. Directly computing $I(U; X_{atk})$ requires the intractable marginal $q_\alpha(U)$. We introduce a tractable reference distribution $r(U) = \mathcal{N}(0, I)$ to obtain a variational upper bound (Alemi et al., 2016). For the d-dimensional Gaussian sampling rule below with fixed $0 < \alpha \leq 1$:

$$\begin{aligned} I(U; X_{atk}) &\leq \mathbb{E}_Z D_{KL}(q_\alpha(U \mid Z) \| r(U)) \\ &= \mathcal{L}_{KL} - d \log \alpha - \frac{1 - \alpha^2}{2} \mathbb{E}_Z \|\sigma(Z)\|^2 \leq \mathcal{L}_{KL} - d \log \alpha, \\ \mathcal{L}_{KL} &= \frac{1}{2} \mathbb{E} \sum_{j=1}^{d} \left(\mu_j^2 + \sigma_j^2 - 1 - \log \sigma_j^2 \right). \end{aligned} \tag{8}$$

Combining message reconstruction with this penalty gives the objective $\mathcal{L}_{rec} + \beta \mathcal{L}_{KL}$.

4.2 Stochastic Bottleneck Implementation

The **stochastic layer** maps backbone features $Z = E_{det}(X_{atk})$ to U. Together with the message prediction head, it forms the VIB decoding head (Figure 2).

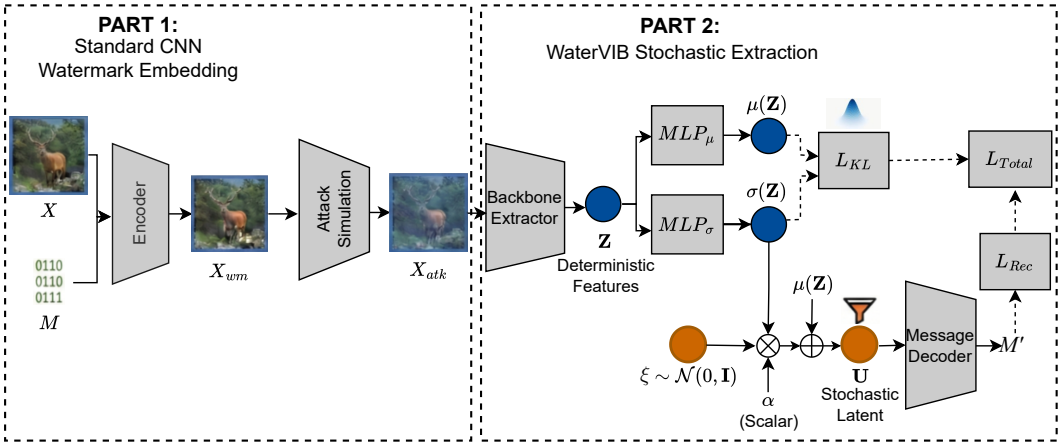
Two parallel MLPs predict the distribution parameters $\mu(Z) = \mathrm{MLP}_\mu(Z)$ and $\sigma(Z) = \exp(\frac{1}{2}\mathrm{MLP}_\sigma(Z))$. Reparameterization (Kingma & Welling, 2013) permits backpropagation through sampling; α scales the stochastic perturbation:

$$U = \mu(Z) + \alpha \cdot \xi \odot \sigma(Z), \quad \xi \sim \mathcal{N}(0, I) \tag{9}$$

Gradients flow through μ and σ, while ξ carries the stochasticity. The message prediction head decodes M' solely from U, so message recovery relies on the bottleneck representation.

Training vs. Inference. During training, sampling U encourages message recovery under stochastic variations in the latent representation. At inference, we use the posterior mean $U = \mu(Z)$ to obtain deterministic predictions without sampling variability.

Figure 2. WaterVIB. After encoding and attack simulation, a stochastic bottleneck compresses features for message decoding. Reconstruction and KL objectives jointly train the embedding and extraction networks.



5 Evaluation

5.1 Unified Experimental Setup

We pair **HiDDeN** (Zhu et al., 2018), **CRMark** (Chen et al., 2025), **Robust-Wide** (Hu et al., 2024), **EditGuard** (Zhang et al., 2024), and **VINE-R** (Lu et al., 2025) with WaterVIB. Each baseline–WaterVIB pair uses the same training settings and budget and is evaluated under the same 32-attack protocol. The matrix spans conventional, geometric, adversarial, local and global generative, reconstruction, compression, and compound processing. We report BER (%) and embedding PSNR (dB). Full train-eval configurations appear in Appendix G.

We independently vary messages and host images on HiDDeN to connect representation-level information with recovery by the original decoder. The base study crosses 512 hosts with 128 messages, measuring backbone and decoding-head features under clean and attacked conditions. We combine complementary information-theoretic estimators, variance decomposition, and kernel analyses to characterize message information, host dependence, and decoded information (Appendix J).

5.2 Message Recovery across Backbones

Across six global purification conditions, WaterVIB reduces BER by an average of 54.2% across five backbones (Table 3(d)). These gains span different architectures, including those built on pretrained generative encoders. HiDDeN and VINE-R also improve embedding PSNR, yielding joint gains in robustness and embedding fidelity.

Figure 3. Message stability and gradient interference. (a,b) Clean (circles) and edited (triangles) message features. (c) WaterVIB reduces the gradient interference ratio by 73%.

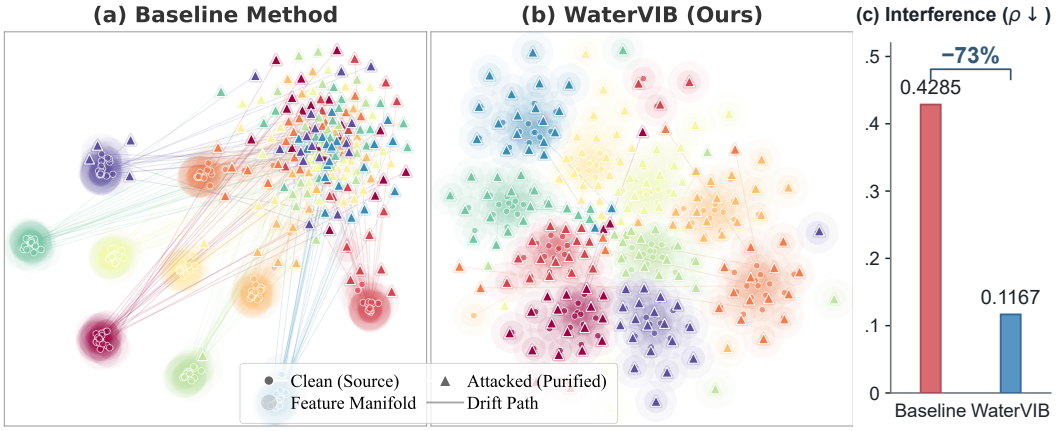


Table 3. WaterVIB under a unified evaluation protocol. BER (%) and PSNR (dB); bold marks the lower BER within each pair. EditGuard's resize and blur results are analyzed in Appendix G.2.

(a) Conventional processing

Backbone	PSNR↑	Clean	JPEG	WebP	GN	Blur	Bright	Poisson	Compound
HiDDeN	36.93	10.277	37.700	41.060	14.843	13.930	10.853	10.333	45.470
+VIB	37.19	**8.770**	**16.037**	**19.750**	**8.943**	**9.010**	**8.877**	**8.713**	**41.480**
CRMark	40.29	0.830	0.950	1.197	0.889	0.844	2.402	0.834	2.628
+VIB	39.26	**0.139**	**0.155**	**0.195**	**0.159**	**0.139**	**1.400**	**0.144**	**1.181**
Robust-Wide	40.41	0.000	0.005	0.031	0.014	0.005	**0.669**	0.000	0.667
+VIB	40.41	0.000	0.005	**0.028**	**0.006**	**0.003**	0.675	0.000	**0.658**
EditGuard	40.4	0.008	23.381	40.895	1.350	46.974	0.122	0.006	48.982
+VIB	40.3	**0.001**	**3.932**	**12.141**	**0.032**	**0.001**	**0.102**	**0.001**	**28.608**
VINE-R	36.00	**0.024**	0.027	0.053	0.042	0.038	**0.157**	**0.024**	0.269
+VIB	36.73	0.028	**0.026**	**0.030**	**0.025**	**0.026**	0.191	0.028	**0.106**

(b) Geometry and missing information

Backbone	Resize	Crop	Rot.	Trans.	Shear	Scaling	Cropout	Dropout
HiDDeN	11.440	22.090	21.417	17.470	11.057	16.817	34.460	28.453
+VIB	**8.907**	**9.003**	**13.243**	**8.827**	**8.773**	**8.790**	**11.320**	**10.947**
CRMark	0.842	50.103	**50.616**	41.136	3.228	5.105	43.131	9.925
+VIB	**0.144**	**49.322**	51.227	**26.313**	**0.298**	**1.231**	**43.069**	**7.006**
Robust-Wide	0.003	**49.813**	**50.317**	**47.673**	28.264	27.558	40.434	0.766
+VIB	0.003	49.959	50.392	47.688	**28.125**	**27.348**	**40.311**	**0.638**
EditGuard	92.507	50.113	50.093	**50.745**	46.779	47.436	28.691	10.163
+VIB	**0.001**	**49.926**	**49.983**	51.290	**10.147**	**30.232**	**28.194**	**1.633**
VINE-R	0.032	**49.877**	**49.888**	51.036	37.529	38.361	**49.719**	7.434
+VIB	**0.031**	49.948	49.893	**50.775**	**35.848**	**35.393**	49.790	**7.059**

(c) Generative editing and reconstruction

Backbone	SD2	CN	SDXL	RePaint	IP2P	VAE	Codec
HiDDeN	10.380	10.346	10.397	10.353	46.806	46.783	45.153
+VIB	**8.793**	**8.797**	**8.817**	**8.780**	**40.283**	**29.827**	**34.653**
CRMark	4.105	3.820	3.544	3.748	18.574	8.850	3.977
+VIB	**3.650**	**3.314**	**2.863**	**3.158**	**10.174**	**1.456**	**0.442**
Robust-Wide	5.663	5.760	5.061	5.916	1.045	0.742	3.148
+VIB	**5.445**	**5.576**	**4.880**	**5.842**	**1.023**	**0.552**	**3.088**
EditGuard	0.055	0.035	0.037	0.028	49.195	48.979	48.473
+VIB	**0.002**	**0.003**	**0.003**	**0.008**	**35.930**	**31.130**	**46.910**
VINE-R	0.080	0.063	**0.059**	**0.058**	5.029	4.344	0.481
+VIB	**0.074**	**0.062**	0.064	0.061	**2.782**	**1.053**	**0.086**

(d) Global purification and specialized attacks

Backbone	SD1.5	SD2	CN	DDPM	SDXL	SDXL-I	Geom.	YUV	DWT	PGD
HiDDeN	46.858	43.907	47.093	31.640	33.860	34.187	16.860	11.497	12.000	83.483
+VIB	**30.620**	**32.933**	**30.608**	**12.790**	**15.773**	**15.883**	**9.570**	**8.863**	**8.937**	**38.833**
CRMark	9.231	5.378	9.385	0.945	1.148	1.153	47.619	1.250	0.841	37.505
+VIB	**1.526**	**0.500**	**1.588**	**0.163**	**0.188**	**0.183**	**39.094**	**0.166**	**0.138**	**17.453**
Robust-Wide	0.915	0.220	0.826	0.011	0.002	0.003	48.956	0.000	0.000	**44.447**
+VIB	**0.743**	**0.181**	**0.663**	**0.009**	**0.000**	**0.002**	**48.786**	0.000	0.000	46.741
EditGuard	48.983	48.608	49.188	60.835	22.997	22.435	50.047	**0.085**	3.281	28.022
+VIB	**31.567**	**29.744**	**31.765**	**2.431**	**9.325**	**8.376**	**49.964**	0.241	**0.004**	**0.593**
VINE-R	5.079	1.573	4.938	0.049	0.086	0.088	50.200	0.024	0.060	7.768
+VIB	**1.475**	**0.507**	**1.424**	**0.030**	**0.041**	**0.041**	**50.137**	0.024	**0.040**	**3.657**

5.3 Message Retention and Host Dependence

WaterVIB preserves message clusters and reduces gradient interference on EditGuard (Figure 3). To evaluate message retention and host dependence jointly, we independently vary messages and host images on HiDDeN. Neural DV estimates (Belghazi et al., 2018) quantify message identity $I(Z; M)$ and conditional host identity $I(Z; C \mid M)$ at the shared 64-dimensional backbone interface. Here, M and C index the crossed messages and host images. We compare these estimates with the bitwise information recovered by the original decoder.

Figure 4. Message identity, host dependence, and decoded information. (a–c) Paired HiDDeN measurements on a 112×64 grid. DV error bars show SD across three critic seeds; decoded-MI error bars show two-way bootstrap 95% intervals. Vertical dashed lines mark index entropies. SD1.5, SD2 and ControlNet are global; SDXL and RePaint are local. (d) Matrix-based Rényi-2 functionals for message and unconditional host labels on a 512×128 grid at bandwidth factor 1; error bars show SD across 32 balanced blocks.

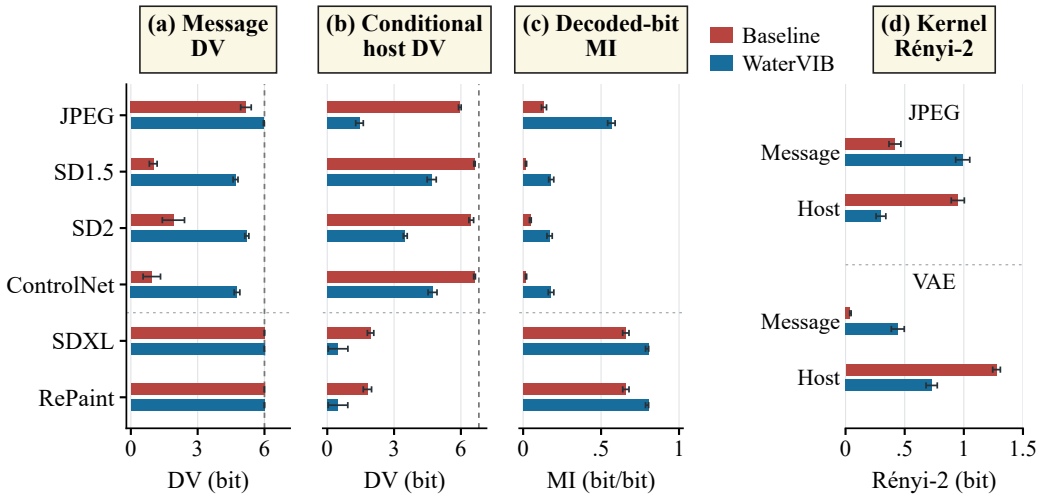
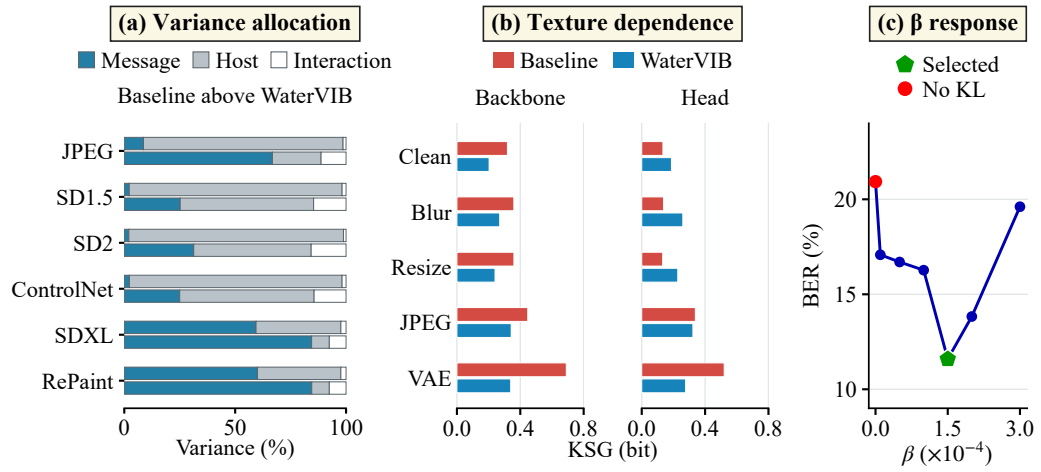


Figure 5. Feature organization and regularization. (a) Message, host, and interaction variance fractions on the grid in Figure 4(a–c). (b) Host-texture dependence of backbone and decoding-head features conditioned on the message, estimated using KSG; lower values indicate weaker dependence. (c) HiDDeN KL-weight sweep with fixed decoder capacity; green marks the selected weight.



Information under generative editing. Across global SD1.5, SD2, and ControlNet attacks, WaterVIB increases the mean message DV estimate by approximately 280% while reducing the mean conditional host DV estimate by 34.4% (Figure 4(a,b)). Under the same attacks, the original decoder's mean bitwise MI increases from 0.028 to 0.175 bit/bit (Figure 4(c)). These gains extend from message identification in the representation to message recovery by the original decoder.

Stable message readout. Classifiers trained only on clean features retain 91.6%–94.6% message-identification accuracy after JPEG compression across WaterVIB's two feature interfaces, compared

with 12.9%–13.3% for the baseline (J.4). The clean-trained readout therefore remains effective after compression without refitting, indicating stable message representations in WaterVIB.

Message and host components. Across the three global attacks, WaterVIB raises the average message-related variance share from 2.2% to 27.2% and reduces the host-related share from 96.2% to 57.9% (Figure 5(a)). This redistribution makes message-related variation more prominent while reducing host-related residuals in the features. Rényi-2 measurements under JPEG and VAE likewise show stronger message structure and less residual host information (Figure 4(d)). Under VAE reconstruction, lower conditional texture KSG persists across both feature interfaces, sample grids, and estimator settings (Figure 5(b)), with consistent support from kernel alignment measurements (Appendix J.6). We further reproduce the recovery gains on additional data (Appendix J.7).

Regularization strength. We sweep β with the expanded decoder architecture fixed (Figure 5(c)); $\beta = 0$ removes only KL regularization. Moderate regularization lowers BER relative to this control, showing gains beyond expansion alone. We select $\beta = 1.5 \times 10^{-4}$.

5.4 Beyond 2D: NeRF Watermarking

We extend WaterVIB to 3D watermarking through the extraction head of NeRF-Signature (Luo et al., 2025). Rendering fidelity improves on both Blender and LLFF while clean decoding is preserved (Table 4). Pixelwise and perceptual metrics agree on this improvement across synthetic and real scenes. The weaker watermark residuals in Figure 6 illustrate this joint improvement.

Across all five tested payloads from 16 to 96 bits on Blender, both PSNR and LPIPS improve (Appendix I.1). PSNR gains range from 0.76 to 3.65 dB, with larger gains at 16–64 bits than at 80–96 bits. Appendix D gives the full 3D protocol.

Figure 6. NeRF renderings and watermark residuals. The residual color scale is shared.

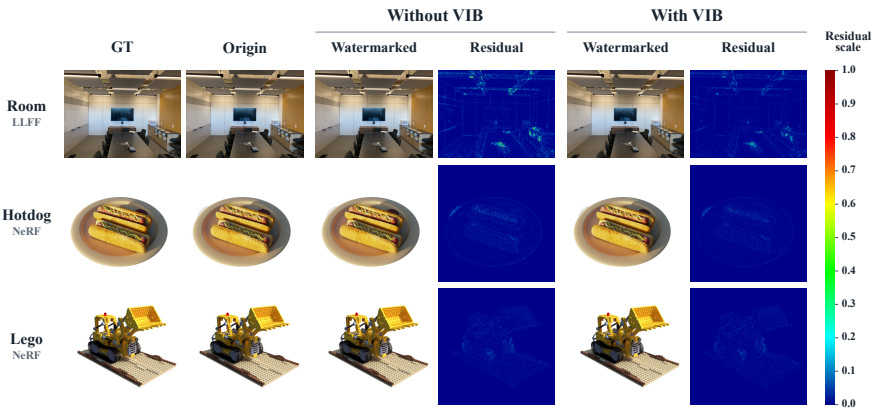


Table 4. NeRF-Signature with and without WaterVIB (32 bits).

	Blender				LLFF			
Method	BER↓	PSNR↑	LPIPS↓	SSIM↑	BER↓	PSNR↑	LPIPS↓	SSIM↑
NeRF-Sig	0.00	57.17	5×10^{-5}	.9998	0.10	46.51	.0016	.9973
+WaterVIB	**0.00**	**58.99**	$\mathbf{3 \times 10^{-5}}$	**.9999**	**0.10**	**49.61**	**.0009**	**.9984**

6 Conclusion

Our findings highlight dependence on host-specific textures as a source of watermark vulnerability to generative editing. This motivates learning representations that retain message information while reducing reliance on image details that editing can rewrite. WaterVIB turns this principle into a plug-in training module through a variational information bottleneck. Its robustness gains across diverse watermarking architectures support message-preserving representation compression as a practical direction for robust watermark design. Future work will explore adaptive bottleneck selection, higher-capacity watermarking, and robustness to adaptive attacks.

IMPACT STATEMENT

WaterVIB supports ownership verification and content provenance when combined with appropriate records and access controls. Applications should protect creators while respecting privacy, consent, and legitimate image editing.

REPRODUCIBILITY STATEMENT

Section 4 specifies the variational objective and module; Appendices A–C provide derivations and architecture details. Appendix G records the common evaluation protocol and model configurations. Our code is available in an anonymous repository: `https://anonymous.4open.science/r/WaterVIB-code-7217/README.md`.

AI USE STATEMENT

The main research work was carried out by the authors. AI assistance began in September 2026 and supported literature lookup, server operations under the authors' direction, and manuscript polishing. AI's fast analytical capabilities also helped improve and apply code and analyze experimental results.

REFERENCES

Achille, A. and Soatto, S. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19(50):1–34, 2018.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *ICML*, volume 80, pp. 531–540, 2018.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Brooks, T., Holynski, A., and Efros, A. A. InstructPix2Pix: Learning to follow image editing instructions. In *CVPR*, pp. 18392–18402, 2023.

Bui, T., Agarwal, S., and Collomosse, J. Trustmark: Universal watermarking for arbitrary resolution images. *arXiv preprint arXiv:2311.18297*, 2023.

Cheng, Z., Sun, H., Takeuchi, M., and Katto, J. Learned image compression with discretized Gaussian mixture likelihoods and attention modules. In *CVPR*, 2020.

Chen, J., Wang, W., Shi, C., Dong, L., and Hu, X. Learning robust image watermarking with lossless cover recovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Field, D. J. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379–2394, 1987. doi: 10.1364/JOSAA.4.002379.

Ganic, E. and Eskicioglu, A. M. Robust dwt-svd domain image watermarking: embedding data in all frequencies. In *MM and Sec Workshop*, pp. 166–174, 2004.

Gilad-Bachrach, R., Navot, A., and Tishby, N. An information theoretic tradeoff between complexity and accuracy. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*, pp. 595–609. Springer, 2003.

Hu, R., Zhang, J., Xu, T., Li, J., and Zhang, T. Robust-Wide: Robust watermarking against instruction-driven image editing. In *European Conference on Computer Vision*, 2024.

Jing, J., Deng, X., Xu, M., Wang, J., and Guan, Z. Hinet: Deep image hiding by invertible network. In *CVPR*, pp. 4733–4742, 2021.

Kang, X., Huang, J., Shi, Y. Q., and Lin, Y. A dwt-dft composite watermarking scheme robust to both affine transform and jpeg compression. *TCSVT*, 13(8):776–786, 2003.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *ICML*, volume 97 of *PMLR*, pp. 3519–3529, 2019.

Kraskov, A., Stögbauer, H., and Grassberger, P. Estimating mutual information. *Physical Review E*, 69:066138, 2004. doi: 10.1103/PhysRevE.69.066138.

Langelaar, G. C. and Lagendijk, R. L. Optimal differential energy watermarking of dct encoded images and video. *TIP*, 10(1):148–158, 2001.

Lehmann, E. L. and Scheffé, H. Completeness, similar regions, and unbiased estimation-part i. In *Selected works of EL Lehmann*, pp. 233–268. Springer, 2011.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *ECCV*, pp. 740–755. Springer, 2014.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Van Gool, L. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.

Luo, Z., Rocha, A., Shi, B., Guo, Q., Li, H., and Wan, R. The nerf signature: Codebook-aided watermarking for neural radiance fields. *TPAMI*, 2025.

Lu, S., Zhou, Z., Lu, J., Zhu, Y., and Kong, A. Robust watermarking using generative priors against image editing: From benchmarking to advances. In *ICLR*, 2025.

Mildenhall, B., Srinivasan, P. P., Ortiz-Cayon, R., Kalantari, N. K., Ramamoorthi, R., Ng, R., and Kar, A. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (ToG)*, 38(4):1–14, 2019.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65 (1):99–106, 2021.

Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., and Anandkumar, A. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*, 2022.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Poole, B., Ozair, S., van den Oord, A., Alemi, A., and Tucker, G. On variational bounds of mutual information. In *ICML*, pp. 5171–5180, 2019.

Razeghi, B., Rezaeifar, S., Ferdowsi, S., Holotyak, T., and Voloshynovskiy, S. Compressed data sharing based on information bottleneck model. In *ICASSP*, pp. 3009–3013. IEEE, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10684–10695, 2022.

Ross, B. C. Mutual information between discrete and continuous data sets. *PLOS ONE*, 9(2):e87357, 2014. doi: 10.1371/journal.pone.0087357.

Sanchez Giraldo, L. G., Rao, M., and Principe, J. C. Measures of entropy from data using infinitely divisible kernels. *IEEE Transactions on Information Theory*, 61(1):535–548, 2015. doi: 10.1109/TIT.2014.2370058.

Sander, T., Fernandez, P., Durmus, A., Furon, T., and Douze, M. Watermark anything with localized messages. In *ICLR*, 2025.

Seo, S., Kim, S., and Park, C. Interpretable prototype-based graph information bottleneck. *NeurIPS*, 36:76737–76748, 2023.

Tancik, M., Mildenhall, B., and Ng, R. Stegastamp: Invisible hyperlinks in physical photographs. In *CVPR*, pp. 2117–2126, 2020.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Yang, Z., Qi, Z., Ren, Z., Jia, Z., Sun, H., Zhu, X., and Liao, X. Exploring information processing in large language models: Insights from information bottleneck theory. *arXiv preprint arXiv:2501.00999*, 2025.

Zhang, C., Benz, P., Karjauv, A., Sun, G., and Kweon, I. S. Udh: Universal deep hiding for steganography, watermarking, and light field messaging. *Advances in Neural Information Processing Systems*, 33:10223–10234, 2020.

Zhang, L., Rao, A., and Agrawala, M. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

Zhang, X., Li, R., Yu, J., Xu, Y., Li, W., and Zhang, J. Editguard: Versatile image watermarking for tamper localization and copyright protection. In *CVPR*, pp. 11964–11974, 2024.

Zhao, X., Zhang, K., Su, Z., Vasan, S., Grishchenko, I., Kruegel, C., Vigna, G., Wang, Y.-X., and Li, L. Invisible image watermarks are provably removable using generative ai. *Advances in neural information processing systems*, 37:8643–8672, 2024.

Zhong, R., Wang, R., Yao, W., Hu, M., Dong, S., and Munteanu, A. Semantic representation and attention alignment for graph information bottleneck in video summarization. *TIP*, 32:4170–4184, 2023.

Zhu, J., Kaplan, R., Johnson, J., and Fei-Fei, L. Hidden: Hiding data with deep networks. In *ECCV*, pp. 657–672, 2018.

A Theoretical Equivalence to Classical Minimal Sufficient Statistics

We establish the equivalence between classical minimal sufficiency Lehmann & Scheffé (2011) and the information conditions in Theorem 3.3 for deterministic statistics on finite domains.

A.1 Preliminaries

Write $X \equiv X_{atk}$ for the extractor observation, with $X_{atk} = X_{wm}$ under the identity attack. The image and message alphabets are finite, and each message in the support has positive probability:

- **Message (Target):** $M \sim p(m)$, representing the discrete watermark message.
- **Observation (Data):** $X \sim p(x|m)$, representing the digitized image received by the extractor.
- **Statistic (Representation):** Let T be a statistic of X, defined as a deterministic function $T = \tau(X)$.

We assume the data generation and feature extraction process follows a Markov Chain:

$$M \to X \to T \tag{10}$$

A.2 Classical Statistical Definitions

We first restate the foundational definitions from classical mathematical statistics adapted for discrete variables.

Definition 1 (Sufficiency). A statistic $\tau(X)$ is sufficient for M if the conditional distribution of X given $\tau(X)$ is independent of M. That is, for all x, t, m:

$$p(x|t, m) = p(x|t) \tag{11}$$

Definition 2 (Minimal Sufficiency via Lehmann-Scheffé). A sufficient statistic $\tau(X)$ is called a Minimal Sufficient Statistic (MSS) if, for any other sufficient statistic $S(X)$, $\tau(X)$ is a function of $S(X)$. That is, there exists a function f such that:

$$\tau(X) = f(S(X)) \quad \text{almost surely.} \tag{12}$$

The MSS induces the coarsest sufficient partition. On finite domains it exists: group observations with identical posteriors $p(M \mid X = x)$.

A.3 Information-Theoretic Foundations

To prove the main theorems, we first establish precise properties of deterministic functions and sufficiency using entropy and mutual information definitions for discrete variables.

Proposition 1 (Vanishing Conditional Entropy for Deterministic Functions). *For any discrete random variable X and a deterministic function $\tau(\cdot)$, the conditional entropy $H(\tau(X)|X)$ is zero.*

Proof. Let $T = \tau(X)$. By the definition of conditional entropy for discrete variables:

$$H(T|X) = -\sum_{x \in \mathcal{X}} p(x) \sum_{t \in \mathcal{T}} p(t|x) \log p(t|x) \tag{13}$$

Since τ is deterministic, the conditional probability mass function $p(t|x)$ is an indicator function: it equals 1 if $t = \tau(x)$ and 0 otherwise. The entropy of a deterministic event is zero ($-1 \log 1 = 0$). Thus, the inner sum vanishes for all x:

$$\sum_{t \in \mathcal{T}} p(t|x) \log p(t|x) = 0 \tag{14}$$

Averaging over $p(x)$ preserves this zero value, yielding $H(T|X) = 0$. □

Proposition 2 (Equivalence of Conditional Independence and Zero Information). *The random variables X and M are conditionally independent given $\tau(X)$ if and only if the conditional mutual information $I(M; X|\tau(X))$ is zero.*

Proof. The conditional mutual information is defined as the expected Kullback-Leibler (KL) divergence:

$$I(M; X|T) = \mathbb{E}_t \left[D_{KL}(p(m, x|t) \| p(m|t)p(x|t)) \right] \tag{15}$$

Direction ($\Rightarrow$): If X and M are conditionally independent given T, then $p(m, x|t) = p(m|t)p(x|t)$. The KL divergence between two identical distributions is zero, thus the term vanishes.

Direction ($\Leftarrow$): A fundamental property of KL divergence (Gibbs' Inequality) states that $D_{KL}(P\|Q) \geq 0$, with equality if and only if $P = Q$. Therefore, if $I(M; X|T) = 0$, it implies $p(m, x|t) = p(m|t)p(x|t)$ for all (m, x, t) with non-zero probability. This factorization is the definition of conditional independence. □

Theorem 1 (Information-Theoretic Criterion for Sufficiency). *A statistic $\tau(X)$ is sufficient for M if and only if it preserves the mutual information between the input and the target:*

$$I(M; \tau(X)) = I(M; X) \tag{16}$$

Proof. We analyze the joint mutual information $I(M; X, \tau(X))$ using the Chain Rule. Since $\tau(X)$ is deterministic given X, $H(\tau(X)|X) = 0$ (Proposition 1), which implies $I(M; \tau(X)|X) = 0$. Thus:

$$I(M; X, \tau(X)) = I(M; X) \tag{17}$$

Alternatively, expanding in the reverse order:

$$I(M; X, \tau(X)) = I(M; \tau(X)) + I(M; X|\tau(X)) \tag{18}$$

Combining these, we get the fundamental identity:

$$I(M; X) = I(M; \tau(X)) + I(M; X|\tau(X)) \tag{19}$$

From this identity, the equality $I(M; X) = I(M; \tau(X))$ holds if and only if $I(M; X|\tau(X)) = 0$. By **Proposition 2**, $I(M; X|\tau(X)) = 0$ is equivalent to conditional independence ($M \perp X|\tau(X)$), which is precisely the definition of Sufficiency. □

Lemma 1 (Information Invariance under Deterministic Mapping). *Let T be a discrete random variable and $Y = g(T)$ be a deterministic function of T. The mutual information with any source X is preserved in the joint pair (T, Y): $I(X; T) = I(X; T, g(T))$.*

Proof. Applying the Chain Rule: $I(X; T, Y) = I(X; T) + I(X; Y|T)$. Since Y is deterministic given T, $H(Y|T) = 0$, implying $I(X; Y|T) = 0$. Thus, $I(X; T, g(T)) = I(X; T)$. □

Lemma 2 (Invertibility from Information Equality). *Let T be a random variable determined by X. Let $T^* = g(T)$. If $I(X; T) = I(X; T^*)$, then g is invertible on the support of T.*

Proof. Define information loss $\Delta I = I(X; T) - I(X; T^*)$. By **Lemma 1**, $I(X; T) = I(X; T, T^*) = I(X; T^*) + I(X; T|T^*)$. Thus, $\Delta I = I(X; T|T^*)$. Given $I(X; T) = I(X; T^*)$, we have $I(X; T|T^*) = 0$. Written as expected KL divergence:

$$\mathbb{E}_{t^*} \left[D_{KL}(p(x, t|t^*) \| p(x|t^*)p(t|t^*)) \right] = 0 \tag{20}$$

By Gibbs' Inequality, this implies $p(x, t|t^*) = p(x|t^*)p(t|t^*)$. Since T is a deterministic function of X (with $T = \tau(X)$), the conditional PMF is $p(t|x, t^*) = \mathbb{I}(t = \tau(x))$, where $\mathbb{I}$ is the indicator function. The conditional independence implies $p(t|x, t^*) = p(t|t^*)$. Therefore, $p(t|t^*) = \mathbb{I}(t = \tau(x))$, meaning T is fully determined by T^*. This proves the existence of an inverse mapping $T = f(T^*)$, making g bijective on the support. □

A.4 Theorem 2: Classical MSS Implies Information Minimality

Theorem 2. *Let $\tau(X)$ be a Minimal Sufficient Statistic for M in the classical sense. Let $S(X)$ be any other sufficient statistic for M. Then, $\tau(X)$ minimizes the mutual information with the input X among all sufficient statistics:*

$$I(\tau(X); X) \leq I(S(X); X) \tag{21}$$

Proof. Let $\tau(X)$ be the Minimal Sufficient Statistic and $S(X)$ be an arbitrary sufficient statistic for M. By the definition of minimal sufficiency (Definition 2), $\tau(X)$ must be a function of $S(X)$, denoted as $\tau(X) = f(S(X))$. This functional relationship establishes a Markov Chain $X \to S(X) \to \tau(X)$.

Applying the Data Processing Inequality (DPI) to this chain, we obtain $I(X;\tau(X)) \leq I(X;S(X))$. Since $\tau(X)$ is sufficient, it retains all mutual information regarding the target M (**Theorem 1**), while the DPI confirms it simultaneously minimizes the information retained about the input X compared to any other sufficient statistic $S(X)$. ■

A.5 Theorem 3: Information Minimality Implies Classical MSS

Theorem 3 (Converse). *Let T be a sufficient statistic for M such that for any other sufficient statistic $S(X)$, $I(T;X) \leq I(S;X)$. Let $T^*(X)$ be a classical Minimal Sufficient Statistic. Then, T is isomorphic to $T^*(X)$.*

Proof. Assume a classical MSS $T^*(X)$ exists. Since T is hypothesized to be a sufficient statistic, by the definition of classical minimal sufficiency, $T^*(X)$ must be a function of T. Let this mapping be $T^*(X) = g(T)$. This dependency forms the Markov Chain $X \to T \to T^*(X)$, and by the Data Processing Inequality, we have $I(X;T^*) \leq I(X;T)$.

Conversely, the theorem hypothesis states that T minimizes mutual information among *all* sufficient statistics. Since $T^*(X)$ is itself a sufficient statistic, it must hold that $I(X;T) \leq I(X;T^*)$. Combining these two inequalities, we obtain the equality:

$$I(X;T) = I(X;T^*) = I(X;g(T)) \tag{22}$$

Invoking **Lemma 2**, the information equality $I(X;T) = I(X;g(T))$ implies that the function g is invertible on the support of T. Consequently, there exists a deterministic inverse $T = g^{-1}(T^*)$. Since T^* is a function of T and T is a function of T^*, T is strictly equivalent to the classical MSS T^* up to isomorphism. ■

B Message-Preserving Compression and Host Dependence

We connect the approximate MSS constraint to the IB objective, characterize its control of host information, and establish the effect of the compression weight.

B.1 Message Sufficiency as Posterior Preservation

Let $X \equiv X_{atk}$ and $C \equiv X_{host}$. Fix $p(c,m,x)$ on finite alphabets and consider stochastic representations with joint distribution $p(c,m,x)q(t \mid x)$. This gives $(C,M) \to X \to T$. The representation class contains all finite output alphabets. Define

$$I_0 = I(X;M) > 0, \qquad R(T) = I(X;T), \qquad J(T) = I(M;T), \qquad \delta(T) = I_0 - J(T).$$

All information quantities in this derivation use natural logarithms.

Proposition B.1 (Posterior preservation). The loss of message information is

$$\delta(T) = I(M;X \mid T) = \mathbb{E}_{X,T} D_{KL}(p(M \mid X) \| p(M \mid T)) \,. \tag{23}$$

Proof. Expand $I(M;X,T)$ in both orders and use $I(M;T \mid X) = 0$. The conditional MI equals the displayed KL expectation since $p(m \mid x,t) = p(m \mid x)$. Thus $\delta(T) \leq \epsilon$ bounds the average change in the message posterior after compression. □

Equivalent rate–distortion formulation. Let a reconstruction $s \in \Delta(\mathcal{M})$ be a message probability vector and set $d(x,s) = D_{KL}(p(M \mid x) \| s)$. The approximate MSS problem has the equivalent value

$$\min_{T:\, \delta(T) \leq \epsilon} I(X;T) = \inf_{q(s|x):\, \mathbb{E}d(X,S) \leq \epsilon} I(X;S). \tag{24}$$

For any feasible T, choose $S = p(M \mid T)$. This preserves $J(T)$, has distortion $\delta(T)$, and cannot increase the rate. Conversely, any S satisfies

$$\mathbb{E}d(X,S) = I_0 - I(M;S) + \mathbb{E}_S D_{KL}(p(M \mid S) \| S) \geq \delta(S).$$

Hence every feasible reconstruction yields a feasible representation. The finite reduction below also covers reconstructions with continuous support. This posterior formulation makes the distortion fixed and its expectation linear in $q(s|x)$, while $I(X;S)$ is convex in that kernel (Tishby et al., 2000).

B.2 Lagrangian Characterization

Define the minimum rate $\mathcal{R}(i) = \min_{T:\, J(T)\geq i} R(T)$ for $0 \leq i \leq I_0$.

Attainment and convexity. For $|\mathcal{X}| = n$, each output induces a posterior $v_t = p(X \mid t)$. The vector

$$\big(v_{t,1}, \ldots, v_{t,n-1}, H(v_t), H(p(M \mid t))\big), \qquad p(m \mid t) = \sum_x p(m \mid x) v_t(x),$$

ranges over a compact set. Its mean fixes $p(X)$, $H(X \mid T)$ and $H(M \mid T)$. By Carathéodory's theorem, at most $n+2$ outputs preserve these means. The achievable information pairs therefore form a compact set, and the minimum is attained. For two encoders, independently choose $Q \in \{1,2\}$ with probabilities $a, 1-a$ and output (Q, T_Q). Both R and J become the corresponding weighted averages. Thus the achievable region and $\mathcal{R}$ are convex.

Constraint activation. If an encoder has $J(T) > i > 0$, retain its output with independent probability $a = i/J(T)$ and otherwise emit a distinct erasure symbol. The new representation has relevance $aJ(T) = i$ and rate $aR(T) < R(T)$. A minimum-rate solution consequently satisfies $J(T^*) = i$. The same construction gives $\mathcal{R}(i_1) \leq (i_1/i_2)\mathcal{R}(i_2) < \mathcal{R}(i_2)$ for $0 < i_1 < i_2 \leq I_0$, since $\mathcal{R}(i_2) \geq i_2$ by data processing.

Theorem B.2 (Supporting multiplier). For each $i = I_0 - \epsilon \in (0, I_0)$, a constrained optimum T^* also maximizes the IB objective for a positive weight β satisfying

$$\beta^{-1} \in \partial\mathcal{R}(I_0 - \epsilon). \tag{25}$$

Proof. A finite convex function has a supporting slope $\lambda \in \partial\mathcal{R}(i)$ at each interior point (Boyd & Vandenberghe, 2004). Since $\mathcal{R}(0) = 0$ and $\mathcal{R}(i) \geq i$, this slope satisfies $\lambda \geq \mathcal{R}(i)/i \geq 1$. For any T,

$$R(T) - \lambda J(T) \geq \mathcal{R}(J(T)) - \lambda J(T) \geq \mathcal{R}(i) - \lambda i.$$

The constrained optimum attains equality. Dividing by $\lambda > 0$ and setting $\beta = 1/\lambda$ gives

$$T^* \in \arg\max_T \{I(T;M) - \beta I(T;X)\}. \tag{26}$$

It also establishes the dual value identity

$$\mathcal{R}(I_0 - \epsilon) = \sup_{\lambda \geq 0} \Big\{ \inf_T \big[R(T) + \lambda\delta(T)\big] - \lambda\epsilon \Big\}. \qquad \square$$

At a differentiable point, Eq. 25 becomes $\beta = 1/\mathcal{R}'(I_0 - \epsilon)$. Affine segments share a supporting weight; a corner admits a range of supporting slopes. On a differentiable, strictly convex interval, the relation is one-to-one between that interval and its image under $1/\mathcal{R}'$.

B.3 Compression and Conditional Host Information

Proposition B.3 (Host-information decomposition). Under $(C, M) \to X \to T$,

$$R(T) = J(T) + I(T;C \mid M) + I(T;X \mid C, M). \tag{27}$$

Proof. The chain rule gives $I(T;X) = I(T;M) + I(T;X \mid M)$. Expanding $I(T;X,C \mid M)$ and using $I(T;C \mid X, M) = 0$ gives the remaining two terms.□ Consequently, every representation with $\delta(T) \leq \epsilon$ satisfies

$$I(T;C \mid M) \leq R(T) - J(T) \leq R(T) - I_0 + \epsilon. \tag{28}$$

At fixed message information, reducing the rate tightens the bound on host information. This connects message-preserving compression to nuisance reduction (Achille & Soatto, 2018) and motivates the paired message and conditional-host measurements in Section 5.3.

For a fixed crossed observation $X = f(C, M)$, the final term in Eq. 27 vanishes, giving

$$J(T) - \beta R(T) = (1 - \beta)I(T; M) - \beta I(T; C \mid M). \tag{29}$$

For $0 < \beta < 1$, this objective rewards message information and penalizes conditional host information. Stochastic observations retain the additional penalty $-\beta I(T; X \mid C, M)$ from Eq. 27. The decomposition applies at any feature interface satisfying the extraction chain.

B.4 Effect of the Compression Weight

Proposition B.4 (Ordered trade-offs). On the same candidate representation class, let T_1, T_2 globally maximize $J - \beta_1 R$ and $J - \beta_2 R$, where $0 < \beta_1 < \beta_2$. Their optimality inequalities imply

$$\beta_1(R_1 - R_2) \le J_1 - J_2 \le \beta_2(R_1 - R_2). \tag{30}$$

Thus $R_2 \le R_1$, $J_2 \le J_1$, and $\delta_2 \ge \delta_1$. If $\beta_2 \le 1$, then

$$(R_1 - J_1) - (R_2 - J_2) \ge (1 - \beta_2)(R_1 - R_2) \ge 0.$$

Increasing the weight therefore reduces input information and, for weights up to one, the input information remaining beyond the message.

Corollary B.5 (Limit of minimal sufficiency). Let $R_s = \mathcal{R}(I_0)$ and let T_β globally maximize the IB objective over the representation class of this appendix. Comparing it with a minimum-rate sufficient representation gives

$$0 \le \delta(T_\beta), \qquad \delta(T_\beta) + \beta R(T_\beta) \le \beta R_s. \tag{31}$$

Hence $\delta(T_\beta) \le \beta R_s$ and $R(T_\beta) \le R_s$. As $\beta \downarrow 0$, relevance tends to I_0. By compactness, every limiting information pair is sufficient and has rate at most R_s; the definition of R_s gives the reverse inequality. Therefore $R(T_\beta) \to R_s$. The objective approaches minimal sufficiency in the information plane. The parameter sweep in Figure 5(c) selects the weight by the trained model's recovery performance.

C Implementation Details

In this section, we provide the parameter-level details of the WaterVIB framework, including the specific network architectures for the Information Sieve mechanism, training hyperparameters, and the configuration of the differentiable noise layers used during the attack simulation phase to ensure reproducibility.

C.1 The WaterVIB Module Design

The WaterVIB module functions as a stochastic information bottleneck inserted between the feature extractor and the message decoder. To ensure numerical stability and gradient flow during end-to-end optimization, we implement the module using the reparameterization trick. Specifically, the latent variable U is computed as

$$U = \mu(Z) + \alpha \cdot \xi \odot \exp(\frac{1}{2}\log\sigma(Z)^2) \tag{32}$$

where $\xi \sim \mathcal{N}(0, I)$ is sampled from a standard normal distribution. The fixed coefficient α scales the sampling noise. To prevent numerical instability such as exploding gradients, we explicitly clip the predicted log-variance $\log\Sigma(Z)^2$ to the range $[-10, 10]$ before the exponential operation.

The training objective is a composite loss function:

$$\mathcal{L}_{total} = \lambda_{img}\mathcal{L}_{img} + \lambda_{rec}\mathcal{L}_{rec} + \beta\mathcal{L}_{KL} \tag{33}$$

The reconstruction loss $\mathcal{L}_{rec}$ is binary cross-entropy between the message and its predicted probabilities. The Gaussian parameter regularizer $\mathcal{L}_{KL}$ is defined in Eq. 8 and weighted by β. The image fidelity loss $\mathcal{L}_{img}$ measures the Mean Squared Error (MSE) between the host and watermarked images.

Information bound for fixed sampling scale. Let d be the latent dimension, $0 < \alpha \leq 1$ fixed, and μ, σ the parameters used in both sampling and Eq. 8. The training channel is $q_\alpha(U \mid Z) = \mathcal{N}(\mu, \alpha^2\,\mathrm{diag}(\sigma^2))$. With $r(U) = \mathcal{N}(0, I)$ and deterministic $Z = E_{det}(X_{atk})$, its average KL satisfies

$$\begin{aligned} K_\alpha &= \mathbb{E}_Z D_{KL}(q_\alpha(U \mid Z)\|r(U)) \\ &= \mathcal{L}_{KL} - \frac{1-\alpha^2}{2}\mathbb{E}\|\sigma\|^2 - d\log\alpha \leq \mathcal{L}_{KL} - d\log\alpha. \end{aligned} \tag{34}$$

The identity follows by substituting the diagonal-Gaussian KL formula. Also,

$$K_\alpha = I(U; X_{atk}) + D_{KL}(q_\alpha(U)\|r(U)),$$

so nonnegativity of KL gives $I(U; X_{atk}) \leq \mathcal{L}_{KL} - d\log\alpha$. For the message prediction distribution $r_\psi(M \mid U)$,

$$\mathcal{L}_{rec} = H(M \mid U) + \mathbb{E}_U D_{KL}(p(M \mid U)\|r_\psi(M \mid U)) \geq H(M \mid U).$$

Combining these bounds yields

$$I(U; M) - \beta I(U; X_{atk}) \geq H(M) - \mathcal{L}_{rec} - \beta\mathcal{L}_{KL} + \beta d\log\alpha. \tag{35}$$

For fixed β, α, d and message distribution, maximizing this lower bound is equivalent to minimizing $\mathcal{L}_{rec} + \beta\mathcal{L}_{KL}$.

C.2 Integration into High-Capacity Architecture (EditGuard)

For the EditGuard backbone, which utilizes a high-capacity encoder-decoder structure, we insert the VIB module after the 16-channel output of the bit decoder to handle the spatial feature maps. The input to the VIB module is a feature map of shape $[B, 16, 400, 400]$. To handle this high dimensionality efficiently, we employ a CNN-based compression pipeline rather than fully connected layers. The features are first downsampled and compressed via a sequence of channel reduction layers: a $Conv2d(16 \to 4)$ layer reduces the channel dimension while downsampling the spatial resolution to 128×128, followed by a $Conv2d(4 \to 2)$ layer. A final convolution layer $Conv2d(2 \to 2)$

produces the distributional parameters μ and $\log \Sigma^2$, each maintaining the shape $[B, 1, 128, 128]$. After sampling the stochastic latent U, the feature map is flattened to a dimension of $16,384$ and projected via a single Linear layer ($16,384 \to 100$) to produce the 100-bit output logits.

Regarding hyperparameters, we trained the model on the COCO2017 dataset at a resolution of 512×512. We set the compression weight β to 0.0003 and the noise scaling factor α to 10^{-4} to mitigate fluctuations during the initial training phase.

C.3 Integration into Lightweight Architecture (HiDDeN)

For the parameter-constrained HiDDeN backbone, the VIB module is integrated between the final convolutional block of the extractor and the linear readout layer. Unlike the CNN-based approach used in EditGuard, this implementation utilizes a Multi-Layer Perceptron (MLP) structure. The global features from the extractor are mapped to a latent dimension of $D = 128$. Two parallel linear layers then map these features to the statistical parameters $\mu \in \mathbb{R}^{128}$ and $\log \Sigma^2 \in \mathbb{R}^{128}$. The sampled latent U is finally mapped to the 30-bit output via a linear readout layer.

This model was trained on 10,000 images from the COCO2014 dataset at a resolution of 256×256. Based on our ablation studies, we determined the optimal compression weight β to be 0.00015 and set the noise scaling factor α to 0.007.

C.4 Differentiable Noise Simulation and Training

To ensure robust generalization, we train the models against a comprehensive suite of differentiable noise layers. Both EditGuard variants are trained with Gaussian noise, Poisson noise, and JPEG compression. Table 5 lists the training distortions for both HiDDeN variants.

Table 5. Differentiable noise settings for HiDDeN training.

Distortion Type	**Parameter**	**Range / Value**
Crop	Ratio	$[0.4, 0.55]$
Cropout	Ratio	$[0.25, 0.35]$
Dropout	Keep Prob.	$[0.65, 0.75]$
Resize (Scaling)	Scale Factor	$[0.4, 0.6]$
JPEG Compression	Coefficients	$(25, 9, 9)$

All models were implemented in PyTorch and trained on NVIDIA 4090 GPUs using the Adam optimizer. The learning rate was initialized at 10^{-3} and adjusted using a scheduler. We used a batch size of 32 for the lightweight HiDDeN architecture and 64 for the high-capacity EditGuard architecture.

C.5 Training Algorithm

We summarize the end-to-end training process of the WaterVIB framework in Algorithm 1. This algorithm details the stochastic information bottleneck mechanism and the optimization of the composite objective function.

Algorithm 1 Training Process of WaterVIB

Input: Training dataset $\mathcal{D}$, Batch size B, Learning rate η, Hyperparameters α, β
Initialize: Encoder E, Extractor E_{ext}, MLPs ($\text{MLP}_\mu, \text{MLP}_\sigma$), Decoder D_{msg} parameters θ
for each training epoch **do**
  for each batch (x, m) in $\mathcal{D}$ **do**
    # 1. Watermark Embedding
    $x_{wm} \leftarrow E(x, m)$ {Generate watermarked image}
    # 2. Attack Simulation (Differentiable)
    $x_{atk} \leftarrow \text{Attack}(x_{wm})$ {Apply noise layers (e.g., Crop, JPEG)}
    # 3. Deterministic Feature Extraction
    $Z \leftarrow E_{ext}(x_{atk})$ {Extract backbone features}
    # 4. Stochastic Bottleneck (VIB)
    $\mu \leftarrow \text{MLP}_\mu(Z)$
    $\sigma \leftarrow \exp\left(\frac{1}{2}\text{MLP}_\sigma(Z)\right)$
    Sample $\xi \sim \mathcal{N}(0, I)$
    $U \leftarrow \mu + \alpha \cdot \xi \odot \sigma$ {Reparameterization Trick (Eq. 32)}
    # 5. Message Decoding
    $\hat{m} \leftarrow D_{msg}(U)$ {Predict message from stochastic latent}
    # 6. Loss Computation
    $\mathcal{L}_{rec} \leftarrow \text{BCE}(m, \hat{m})$ {Reconstruction Loss (Eq. 7)}
    $\mathcal{L}_{KL} \leftarrow D_{KL}(\mathcal{N}(\mu, \sigma^2) \| \mathcal{N}(0, I))$ {Compression Loss (Eq. 8)}
    $\mathcal{L}_{img} \leftarrow \text{MSE}(x, x_{wm})$ {Image Fidelity Loss}
    $\mathcal{L}_{total} \leftarrow \lambda_{img}\mathcal{L}_{img} + \lambda_{rec}\mathcal{L}_{rec} + \beta\mathcal{L}_{KL}$ {Total Objective (Eq. 33)}
    # 7. Optimization
    Update $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_{total}$
  end for
end for

D Appendix D: Experiments on NeRF-Signature

To demonstrate the universality of our method beyond 2D static images, we integrated the WaterVIB module into NeRF-Signature Luo et al. (2025), a state-of-the-art watermarking framework for 3D Neural Radiance Fields (NeRF). This appendix details the experimental setup and the performance gains achieved by our method in the 3D domain.

D.1 Experimental Setup

Datasets and Protocol. Following the baseline NeRF-Signature framework, we evaluated our method on two standard datasets: the synthetic **Blender** dataset Mildenhall et al. (2021) (Lego, Hotdog, etc.) at its native resolution, and the real-world **LLFF** dataset Mildenhall et al. (2019) (Fern, Room, etc.) downsampled to $1/8$ resolution.

Implementation Details. We embed a 32-bit watermark message into the NeRF representation. The baseline NeRF-Signature injects a compact signature codebook into the radiance field. We integrated our WaterVIB module into the extraction head of the NeRF-Signature pipeline to optimize the information flow. To control randomness, all experiments employed identical NeRF cores, and evaluations were conducted on a fixed set of test viewpoints.

Metrics. We evaluate **imperceptibility** by comparing *Watermarked* vs. *Clean NeRF* renderings using PSNR, SSIM, and LPIPS Luo et al. (2025). **Robustness** is measured via Bit Accuracy (ACC) under standard 2D distortions, including Rotation, Scaling, Blurring, Brightness, JPEG, and Cropping.

D.2 Performance Evaluation

Quantitative Results (Imperceptibility). As shown in Table 4, the integration of WaterVIB significantly improves the visual quality of the watermarked renderings compared to the baseline. On the Blender dataset, WaterVIB increases the PSNR from 57.17 dB to 58.99 dB and reduces the LPIPS perceptual distance by 40%, indicating that our method introduces significantly fewer artifacts while maintaining a 0% Bit Error Rate (BER). A similar trend is observed on the real-world LLFF dataset.

Quantitative Results (Robustness). We further evaluated the robustness of the 3D watermark against 2D image distortions. As detailed in Table 6, WaterVIB maintains the robustness of the baseline (ACC $\approx$ 100%) across all noise categories. W

aterVIB significantly improves imperceptibility, consistently delivering higher PSNR scores under all tested distortions (e.g., +1.3 dB in Cropping). This result confirms that the Information Sieve mechanism effectively concentrates the watermark signal into robust yet imperceptible subspaces, thereby enhancing visual fidelity without compromising extraction accuracy.

Table 6. Robustness evaluation on the Blender dataset (32 bits). We report the Bit Accuracy (ACC, ↑) and the PSNR (↑) of the watermarked views. WaterVIB maintains high robustness while consistently improving visual quality.

Method	Rotation		Scaling		Blurring		Brightness		JPEG		Cropping		Average	
	Acc	PSNR	Acc	PSNR	Acc	PSNR	Acc	PSNR	Acc	PSNR	Acc	PSNR	Acc	PSNR
NeRF-Sig	**0.999**	52.36	1.000	54.94	1.000	55.55	1.000	55.20	**0.999**	51.76	1.000	56.23	0.999	54.32
+ WaterVIB	0.997	**53.26**	1.000	**55.76**	**1.000**	**57.07**	1.000	**55.98**	0.998	**51.93**	1.000	**57.54**	0.999	**55.26**

D.3 Qualitative Analysis

Visual comparisons confirm that the WaterVIB module helps in concentrating the watermark signal into texture-rich, high-frequency regions of the scene, where human vision is less sensitive to perturbations. As observed in our qualitative results (Figure 6), the residual difference maps for WaterVIB are sparser and more localized compared to the baseline, which explains the simultaneous improvement in both imperceptibility (higher PSNR) and robustness.

E Analysis Experimental Setup and Supplement

In this section, we explicitly detail the experimental settings used for the theoretical validation provided in Section 3 and present additional supplementary experiments to further substantiate our findings.

E.1 Setup for Theoretical Analysis (Section 3)

The theoretical analysis presented in Section 3, including the *Spectral and Spatial Alignment* (Observation 3.1) and the *Gradient Counter-Optimization* analysis (Section 3.2), was conducted empirically to validate our propositions.

Model Architecture. The signal analyses in Section 3 use EditGuard and HiDDeN. The settings below describe the EditGuard measurements; the cross-architecture results are summarized in Appendix E.3.

Data Sampling. To strictly evaluate the statistical behavior of the watermark under attack, we randomly sampled **500 images** from the COCO validation dataset (consistent with the validation set used for EditGuard). This sample size was chosen to provide a statistically significant estimate of the spectral energy distribution and gradient projections while maintaining computational feasibility for the extensive gradient tracking required in Section 3.2.

AIGC Purification Setting. The EditGuard measurements in Tables 1, 2, and 7 use SDXL 1.0 in image-to-image mode. The global purification settings are specified in Appendix G.

Model Configuration. The EditGuard integration is described in Appendix C; the model configurations are specified in Appendix G.

E.2 Additional Supplementary Analysis Experiments

To provide a more comprehensive analysis of WaterVIB, we conduct the following additional experiments.

Geometric Orthogonality and Energy Analysis. To strictly quantify the disentanglement, we analyze the geometric relationship between the watermark signal $\mathbf{s}_{wm}$ and the AIGC distortion vector $\mathbf{s}_{atk}$ on 500 randomly sampled images.

Table 7. Orthogonality Analysis. We compare the cosine similarity and the effective projection of the attack noise onto the watermark direction. **Effective Proj.** serves as the decisive metric for signal survival.

Method	**Cosine Sim.** $(\cos\theta)$	**Effective Proj.** $(\eta \cdot \lvert\cos\theta\rvert)$
Baseline (EditGuard)	-0.0395	**0.1572**
WaterVIB (Ours)	**0.0077**	**0.0306**

Analysis of Energy Disparity. A seemingly negligible cosine similarity in high-dimensional spaces can be destructive due to the significant energy gap between the signal and the attack. In our experiments, the watermark is imperceptible ($\text{PSNR}_{wm} \approx 40$ dB), while the AIGC purification introduces substantial distortions ($\text{PSNR}_{atk} \approx 28$ dB). This ≈ 12 dB difference implies that the AIGC noise vector is significantly stronger in magnitude than the watermark signal:

$$\eta = \frac{||\mathbf{s}_{atk}||}{||\mathbf{s}_{wm}||} \approx 10^{\frac{40-28}{20}} \approx 3.98 \tag{36}$$

We define the *Effective Projection* as the ratio of the destructive interference to the watermark strength: $E_{proj} = \eta \cdot |\cos\theta|$. For the Baseline, the attack noise exerts a projection force equivalent to $\mathbf{15.7}\%$ (3.98×0.0395) of the watermark's magnitude, effectively overwriting the signal. In contrast, WaterVIB achieves near-perfect orthogonality ($\cos\theta \approx 0.007$), suppressing the effective interference to a negligible $\mathbf{3.0}\%$, thus ensuring robustness despite the overwhelming energy of the generative process.

E.3 Mechanism Analysis across Architectures

Table 1 reports signal analyses for EditGuard and HiDDeN. The EditGuard extraction and projection settings are given in Appendix E.1. HiDDeN's watermark places 70.95% of its energy in the high-frequency band, while its AIGC distortion places 94.86% there. The signed correlations with the host image, .1020 for the watermark and $-.3414$ for editing, identify content dependence; the negative editing correlation indicates opposition to host content. Both architectures exhibit a negative watermark projection and a positive editing projection onto the decoding-loss gradient.

The additional HiDDeN geometric summaries report an interference ratio .3742, projection ratio .1708, and AIGC–watermark cosine similarity $-.1051$. All three summaries retain their reported precision. Disentanglement targets dependence on the host, rather than eliminating all high-frequency watermark energy.

E.4 Feature and Gradient Measurement Details

For Figure 3(a,b), we select ten watermark messages and embed each message into twenty host images. Clean and edited decoder features are projected jointly into the same two-dimensional t-SNE embedding.

The gradient interference ratio in Figure 3(c) is

$$\rho = \frac{\langle s_{atk}, \nabla_x \mathcal{L} \rangle}{|\langle s_{wm}, \nabla_x \mathcal{L} \rangle|}. \tag{37}$$

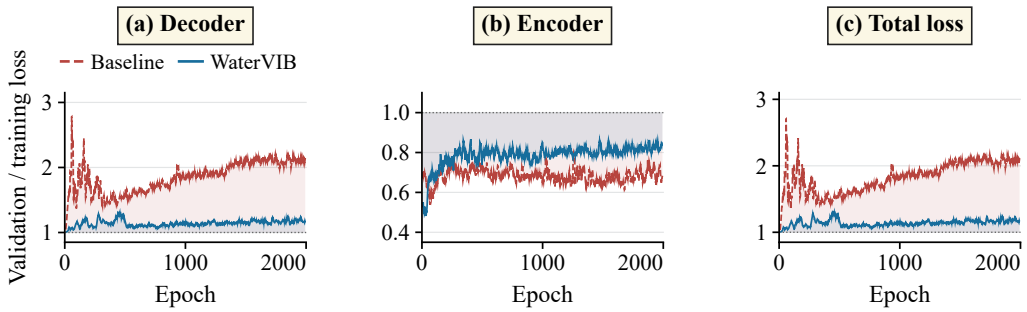
It compares the signed attack projection with the magnitude of the watermark projection onto the decoding-loss gradient.

For Figure 7, we track validation-to-training loss ratios for the decoder, encoder, and total objective in a 2,000-epoch HiDDeN training study. Figure 5(c) reports BER across seven β settings with the expanded decoder architecture fixed; $\beta = 0$ removes KL regularization. This capacity-matched comparison isolates the benefit of KL regularization from architectural expansion. Appendix B characterizes the trade-off between message retention and input compression.

E.5 Training Dynamics

A validation/training loss ratio near one indicates similar losses on the two splits. WaterVIB keeps the decoder and total-objective ratios closer to one throughout training (Figure 7(a,c)). The encoder ratio is higher than the baseline and remains below one (b). These curves complement the KL-weight sweep in Figure 5(c).

Figure 7. HiDDeN training dynamics. Validation/training loss ratios for (a) the decoder, (b) the encoder, and (c) the total objective. One denotes equal losses.



F Defense against Re-Embedding Attacks

In real-world API deployment scenarios, a potential threat arises from *Re-Embedding Attacks* (also known as Multi-Watermarking), where an adversary attempts to overwrite or confuse the existing watermark by embedding a new message into an already watermarked image.

F.1 Defense Mechanism: Pre-Embedding Detection

To mitigate this threat, we implement a **detection-based defense mechanism**. The system enforces a strict policy: *no new embedding is permitted if a watermark is already detected.* This requires the decoder to function not only as a message extractor but also as a robust binary detector (Watermarked vs. Clean).

We leverage the output logits of the decoder as a confidence metric. Specifically, we define the **Average Logits (AL)** score:

$$AL = \frac{1}{L}\sum_{i=1}^{L} |l_i| \tag{38}$$

where l_i represents the raw logit output for the i-th bit of the message. Due to the sigmoid activation in the final layer, higher absolute logits $|l_i|$ correspond to higher confidence predictions (approaching 0 or 1).

F.2 The VIB Advantage in Detection

The WaterVIB framework inherently enhances detectability. The Information Bottleneck objective ($\mathcal{L}_{IB}$) constrains the latent representation to capture only the *Minimal Sufficient Statistics* of the message. The message-reconstruction objective encourages confident predictions for embedded bits.

This distributional separation effectively widens the gap between the two classes, making the detection threshold robust even under noise.

F.3 Empirical Evaluation

We evaluated this defense on a held-out set of 1,500 images: 500 clean images, 500 watermarked images embedded with random messages, and 500 watermarked images subjected to Gaussian noise

($\sigma = 0.05$). We compare the detection performance of our **HiDDeN-VIB** against the standard **HiDDeN** baseline.

Metrics. We report the False Positive Rate (**FP**) on clean images, False Negative Rate (**FN**) on watermarked images, and the **FN** under noise. Additionally, we measure the Kullback-Leibler Divergence (**KL-Div**) between the logit distributions of clean and watermarked samples to quantify separability.

Results. As shown in Table 8, both models achieve perfect separation (0% error) on clean and standard watermarked images. However, under noise attacks, the baseline's detection capability degrades significantly (FN rises to 38.50%). In contrast, WaterVIB maintains robust detection with an FN of only **1.98%**. The higher KL-Divergence (**3.84** vs. 3.33) confirms that WaterVIB learns a more distinct and robust watermark manifold, effectively preventing re-embedding attacks even when the image has been degraded.

Table 8. Re-Embedding Defense Performance. Comparison of detection robustness. The VIB constraint significantly improves detection under noise, preventing unauthorized re-embedding.

Model	**FP** (Clean)	**FN** (Watermarked)	**FN** (WM + Noised)	**KL-Div** ↑
HiDDeN (Baseline)	0.00%	0.00%	38.50%	3.33
HiDDeN-VIB (Ours)	**0.00%**	**0.00%**	**1.98%**	**3.84**

G Unified Evaluation Protocol

Images and messages. Within each baseline/+VIB pair, image IDs, message bits, masks, and random realizations are identical.

Images are converted to RGB and a 512×512 canonical canvas with bicubic interpolation. Each encoder uses its native resolution; the delivery path clips, rounds, and converts the watermarked image to a 512×512 RGB PNG. Before decoding, the image returns to the backbone's native resolution. Fixed per-image message banks supply the native 30-, 64-, or 100-bit messages. Model and hyperparameter selection uses development data, and all configurations are fixed before final evaluation.

Conventional perturbations. JPEG uses quality 70 and 4:2:0 chroma subsampling. WebP uses quality 70 and method 6. Gaussian noise has independent RGB components with $\sigma = .03$, followed by clipping and quantization. Gaussian blur uses $\sigma = 1$, kernel 7, and reflection padding. Brightness multiplies pixels by 0.75 or 1.25 according to the fixed image seed. Resize downsamples to 256 then restores 512 with bicubic interpolation. Crop retains approximately 70% of the area at a seeded random location and restores the canvas. Rotation uses $\pm 10^\circ$, a fixed canvas, bicubic sampling, and reflection padding.

Local generative editing. Masks use nearest-neighbor resizing and threshold 127, with one indicating the edited region. Outputs composite the generated mask region with the current watermarked input outside it. SD2 uses 512 pixels, nominal 50 steps, strength 1, guidance 7.5, and eta 0. ControlNet uses SD1.5 with the v11p inpainting control model, 512 pixels, nominal 20 steps, strength 1, guidance 7.5, eta 1, and control scale .5. SDXL inpainting runs at 1024 pixels, nominal 50 steps, strength .8, guidance 7.5, eta 0, and conditioning target size 512. RePaint uses the CelebA-HQ 256 model followed by restoration to 512, nominal 150 steps, eta 0, jump length 10, and ten jump samples. The first three editors use empty text prompts. RePaint uses per-sample random generators; batch membership and random-state records are retained.

Model configurations. For each baseline architecture, WaterVIB replaces the standard prediction head at the end of the decoder with a VIB decoding head, as described in Section 4. For example, Robust-Wide uses a 128-channel Gaussian bottleneck. All models use FP32 arithmetic, and VIB decoding uses the posterior mean at inference.

Embedding quality. Table 3 reports embedding PSNR at native resolution. EditGuard retains the previously reported values under its original evaluation convention.

Aggregate robustness. For the six global purification conditions in Table 3(d), we first average the reported BER across attacks within each backbone pair. We then compute the relative reduction of this mean and average the five reductions with equal backbone weights.

Table 9. Embedding fidelity and native model settings for Table 3.

Backbone	Variant	Payload	Resolution	PSNR (dB)
HiDDeN	Base	30 bits	256^2	36.93
HiDDeN	+VIB	30 bits	256^2	37.19
CRMark	Base	64 bits	256^2	40.29
CRMark	+VIB	64 bits	256^2	39.26
Robust-Wide	Base	64 bits	512^2	40.41
Robust-Wide	+VIB	64 bits	512^2	40.41
EditGuard	Base	100 bits	512^2	40.4
EditGuard	+VIB	100 bits	512^2	40.3
VINE-R	Base	100 bits	256^2	36.00
VINE-R	+VIB	100 bits	256^2	36.73

Generative outputs. The Stable Diffusion safety checker can filter outputs from ControlNet, InstructPix2Pix, and SD-v1.5; BER is computed only over successfully generated outputs.

G.1 Expanded Attack Conditions

The unified matrix contains 32 attack conditions and clean decoding for each of ten fixed model configurations. The conditions below extend the conventional and local-edit settings above. Each pair is evaluated on the same 1,000 image IDs within its designated bank. The ten sets of encoded delivery PNGs are reused after source and checkpoint hash verification; each new public attack is generated with a shared implementation and then decoded through the model's original inference interface.

Randomness and processing. New stochastic attacks derive their seed from the image ID and attack identifier, excluding the method name. The baseline and VIB versions of each image use the same random variables. Poisson noise uses a photon scale of 10,000 and inverse-CDF sampling with common pixelwise uniform variates. Cropout preserves a random rectangle with .3 of the height and .3 of the width (approximately .09 area); the remaining pixels are replaced by the canonical clean host image. Dropout preserves each spatial location with probability .3, using a channel-shared mask and replacement with the clean host image elsewhere.

Geometry and transforms. Translation independently samples each axis within $\pm.04$ of the canvas size. Shear ranges over $\pm3^\circ$ and scaling over $[.95, 1.05]$. The combined geometric condition also samples rotation within $\pm8^\circ$. These operations retain the canvas and use bilinear interpolation with border padding. YUV processing performs two RGB–YUV round trips, quantizing each channel with step $4/255$ and adding uniform jitter in $[-1.5, 1.5]/255$, halved for Y. DWT uses two-level Haar decomposition with symmetric boundaries, retaining low-frequency coefficients and multiplying detail coefficients by .5. The combined processing chain downsamples 512 to 256 pixels, multiplies brightness by 1.1, applies JPEG Q65 with 4:2:0 subsampling, and restores 512 pixels.

Global regeneration. SD1.5, SD2, ControlNet, and DDPM use strength .002; the SDXL variants use .05. For the image-to-image pipelines, the requested step count starts at 20 and becomes $\lfloor 1/s \rfloor + 1$ when $20s < 1$, where s is strength. SD1.5/SD2/ControlNet operate at 512 pixels, SDXL variants at 1024 followed by restoration, and DDPM at 256 followed by restoration. Inpainting uses a full-white mask; ControlNet uses the current input as its control image. The diffusion pipelines use the prompt "a high-quality photo", guidance 7.5, and eta 0. DDPM sets its starting diffusion index to $\lfloor 1000s \rfloor$ and uses the scheduler indices no larger than this value. Initial and scheduler noise are explicitly seeded. The global conditions retain each pipeline's shipped output checker; checker settings are identical within each pair. SDXL-Inpainting corresponds to the pipeline previously called SDXL-Refiner in the original local-edit code; global and local columns differ in mask support and settings.

Additional image models. VAE reconstruction uses the SD1.5 VAE posterior mode in fp32 for one encode–decode pass. Learned compression uses the MSE-trained Cheng2020-anchor quality-4 model, with actual entropy encoding and decoding on CPU; bitstream hashes and bits per pixel are

recorded. InstructPix2Pix uses revision `31519b5cb02a7fd89b906d88731cd4d6a7bbf88d`, 50 steps, text guidance 7, image guidance 1.5, and eta 0. Image IDs cycle through watercolor, oil painting, pencil sketch, and warm-lighting instructions, with 250 images per instruction and identical instructions across models.

Adversarial evaluation. PGD optimizes each model's own decoder for 40 steps with one seeded random start, step size .001, and ℓ_∞ budget .01 on the $[0, 1]$ delivery image. This corresponds to .02 on the historical $[-1, 1]$ scale. EditGuard and its WaterVIB variant use the same PGD loss function. VIB inference uses the posterior mean. The forward pass retains PNG quantization, Pillow resizing, and internal hard rounding; the backward pass uses straight-through quantization and antialiased bicubic resize gradients. Iterates are selected first by wrong-bit count and then loss. The final serialized image is decoded through the original adapter and checked against the pixel budget. Every image records a finite, nonzero gradient and agreement between the attack's forward path and the inference adapter.

InstructPix2Pix instructions. "Make this image look like a watercolor painting."; "Make this image look like an oil painting."; "Make this image look like a pencil drawing."; "Make the lighting in this image warmer."

G.2 EditGuard under Resizing and Blur

The large EditGuard gaps in Table 3 reflect different decoding responses to resampling and smoothing. We examine the per-image errors for the same 1,000 images and 100-bit messages used in that table. Resize applies bicubic downsampling from 512×512 to 256×256 followed by restoration to 512×512; blur uses a 7×7 Gaussian kernel with $\sigma = 1$. Table 10 shows how errors are distributed across images.

Table 10. EditGuard error distributions. BER, medians, and quartiles are in percent. The last column counts images whose BER exceeds 50%, out of 1,000.

Condition	Method	BER	Median	Quartiles [25, 75]	Images $> 50\%$
Clean	Baseline	0.008	0	[0, 0]	0
	+WaterVIB	0.001	0	[0, 0]	0
Resize	Baseline	92.507	96	[90.75, 99]	992
	+WaterVIB	0.001	0	[0, 0]	0
Blur	Baseline	46.974	46	[24, 70]	465
	+WaterVIB	0.001	0	[0, 0]	0

Resizing induces predominantly reversed predictions. The baseline's 92.507% BER is not driven by a few outliers: 992 images exceed 50% BER, and 773 reach at least 90%. Errors affect both bit values, with 92.703% of true ones decoded as zero and 92.309% of true zeros decoded as one. Predictions remain approximately balanced (49.576% ones), indicating widespread bit reversal rather than collapse to a constant output. Random guessing gives an expected BER of 50%; it is not an upper bound when predictions are negatively associated with the message.

Blur produces heterogeneous decoding failure. Although the baseline's mean BER is close to 50%, its per-image interquartile range is 24–70%. Thus, the aggregate conceals substantial image-to-image variation rather than uniform random guessing. WaterVIB produces only one erroneous bit among 100,000 bits in each of the clean, resize, and blur conditions, with 999 of 1,000 messages recovered exactly in each condition.

Decoder response. For the baseline, the original decoding rule thresholds the raw output at zero. A separate four-image diagnostic gives identical predictions from the original decoder and the evaluation interface. In this diagnostic, the mean raw response for true ones changes from $+4.498$ on clean inputs to -1.590 after resizing; for true zeros it changes from -4.441 to $+1.669$. This sign reversal agrees with the full-set error pattern. All reported BER values retain the original bit labels and decoding rule.

G.3 Purification Strength and Image Fidelity

Low-strength BER is reproduced from Table 3. We additionally evaluate VINE-R and Robust-Wide under SD1.5 and SD2 purification at strengths 0.05, 0.3, and 0.5 on an additional image set, measuring both BER and post-attack PSNR relative to the original host image.

Table 11. Global purification at increased strengths. Entries report baseline / WaterVIB; bold marks the lower mean BER. PSNR compares attacked images with their original clean hosts.

		VINE-R		Robust-Wide	
Editor	Strength	BER (%)	PSNR (dB)	BER (%)	PSNR (dB)
SD1.5	0.002†	5.079 / **1.475**	—	0.915 / **0.743**	—
SD1.5	0.05	4.653 / **2.694**	24.08 / 24.17	1.941 / **1.744**	24.40 / 24.40
SD1.5	0.3	21.598 / **20.941**	19.64 / 19.71	23.108 / **22.895**	19.82 / 19.82
SD1.5	0.5	33.416 / **33.300**	17.16 / 17.22	36.880 / **36.720**	17.30 / 17.30
SD2	0.002†	1.573 / **0.507**	—	0.220 / **0.181**	—
SD2	0.05	4.451 / **2.608**	24.71 / 24.80	1.936 / **1.678**	25.01 / 25.01
SD2	0.3	21.596 / **20.696**	19.95 / 20.01	23.044 / **22.703**	20.12 / 20.12
SD2	0.5	33.414 / **33.292**	17.22 / 17.27	36.703 / **36.611**	17.35 / 17.35

†BER reproduced from Table 3. Other rows report the supplementary sweep.

WaterVIB improves message recovery at comparable image fidelity: baseline and WaterVIB have nearly identical mean post-attack PSNR at each supplementary strength. The BER advantage persists across both backbones and both editors at strength 0.3, with paired-bootstrap 95% confidence intervals excluding zero in all four comparisons. At strength 0.5, the recovery gap narrows as both methods lose decoding accuracy and image fidelity deteriorates further.

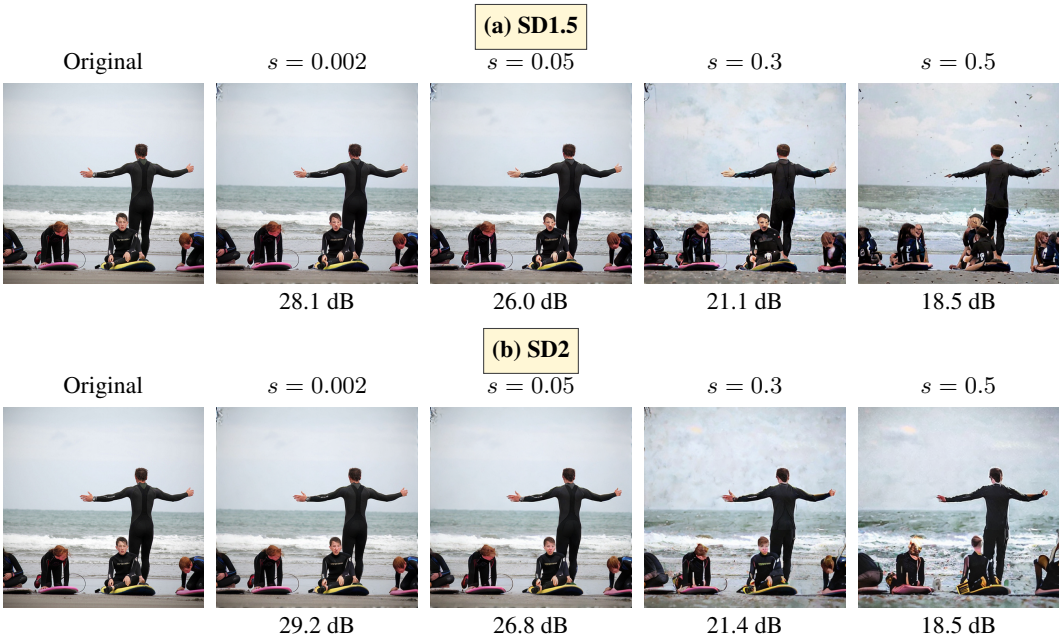


Figure 8. Image changes across purification strengths. One fixed source image with VINE-R + WaterVIB. Labels report PSNR relative to the original host.

Meaningful watermark removal should preserve the visual content that makes the image useful. At $s = 0.5$, Figure 8 shows severely distorted faces and body contours, while Table 11 reports mean post-attack PSNR of approximately 17 dB. Watermark disruption at this strength therefore comes with substantial damage to the image itself. Our main evaluation emphasizes content-preserving removal, using a shared strength of 0.002 for SD1.5, SD2, ControlNet, and DDPM, with identical attack settings within each baseline–WaterVIB pair.

H WaterVIB on VINE-B

Setup. We evaluate VINE-B across all 32 attacks and clean decoding. Both variants retain the native 100-bit payload and 256×256 resolution, with attacks applied to 512×512 delivery images.

Table 12. WaterVIB on VINE-B. BER (%) and PSNR (dB) under the unified protocol; bold marks the lower BER within each pair.

(a) Conventional processing

Backbone	PSNR↑	Clean	JPEG	WebP	GN	Blur	Bright	Poisson	Compound
VINE-B	39.22	0.105	0.131	0.252	0.168	0.186	0.470	0.100	1.193
+VIB	39.90	**0.054**	**0.068**	**0.094**	**0.081**	**0.081**	**0.450**	**0.050**	**0.439**

(b) Geometry and missing information

Backbone	Resize	Crop	Rot.	Trans.	Shear	Scaling	Cropout	Dropout
VINE-B	0.128	**49.944**	**49.897**	50.750	38.753	39.274	**49.787**	11.395
+VIB	**0.060**	49.977	49.929	**50.674**	**36.991**	**36.933**	49.833	**11.132**

(c) Generative editing and reconstruction

Backbone	SD2	CN	SDXL	RePaint	IP2P	VAE	Codec
VINE-B	0.152	0.149	0.147	0.156	8.929	9.958	2.119
+VIB	**0.102**	**0.113**	**0.095**	**0.104**	**5.165**	**3.999**	**0.724**

(d) Global purification and specialized attacks

Backbone	SD1.5	SD2	CN	DDPM	SDXL	SDXL-I	Geom.	YUV	DWT	PGD
VINE-B	11.033	4.778	11.020	0.226	0.400	0.416	50.185	0.116	0.246	11.793
+VIB	**4.968**	**2.239**	**4.668**	**0.082**	**0.173**	**0.189**	**50.127**	**0.053**	**0.090**	**6.521**

Results. VINE-B provides a complementary, higher-fidelity operating point. WaterVIB reduces the mean BER across the 12 AIGC conditions by 53.8% while increasing embedding PSNR from 39.22 to 39.90 dB. Improvements under both local editing and global purification show that the effect extends across distinct generative transformations. The simultaneous fidelity improvement further shows that robustness does not depend on stronger visible embedding.

I Supplementary Architecture Studies

I.1 NeRF Watermarking across Payloads

Table 13 evaluates 16, 32, 64, 80, and 96 bits on Blender. WaterVIB improves PSNR and LPIPS at every tested payload. The payload sweep reports rendering averages separately from the matched-scene 32-bit experiment in Table 4.

Table 13. NeRF-Signature across five payloads on Blender. Bold marks the better value within a pair; ties are unbolded. BER is reported as a fraction.

Bits	Method	PSNR (dB)↑	SSIM↑	LPIPS↓	BER↓
16	Baseline	62.03	.9999	1.489×10^{-5}	0
	+WaterVIB	**65.39**	**1.0000**	$\mathbf{7.250 \times 10^{-6}}$	0
32	Baseline	55.16	.9996	1.480×10^{-4}	1.24×10^{-4}
	+WaterVIB	**58.81**	**.9999**	$\mathbf{3.800 \times 10^{-5}}$	$\mathbf{5.20 \times 10^{-5}}$
64	Baseline	52.39	.9994	1.630×10^{-4}	1.38×10^{-3}
	+WaterVIB	**54.84**	**.9996**	$\mathbf{1.060 \times 10^{-4}}$	$\mathbf{1.27 \times 10^{-4}}$
80	Baseline	53.69	.9995	1.540×10^{-4}	6.17×10^{-3}
	+WaterVIB	**54.45**	**.9996**	$\mathbf{1.110 \times 10^{-4}}$	$\mathbf{1.86 \times 10^{-3}}$
96	Baseline	52.02	.9995	1.560×10^{-4}	$\mathbf{6.50 \times 10^{-6}}$
	+WaterVIB	**52.99**	.9995	$\mathbf{1.470 \times 10^{-4}}$	3.90×10^{-5}

I.2 Computational Cost

Table 14 reports parameter counts for the original integrations. Table 15 compares memory and time per epoch at the stated batch sizes. These paired architecture measurements describe the module overhead.

Table 14. Parameter counts for the original WaterVIB integrations. Counts are in millions.

Backbone	Base	+WaterVIB	Added	Increase
HiDDeN	.4880	.5067	.0187	3.84%
EditGuard	11.4956	12.1347	.6391	5.56%
Robust-Wide	445.9174	445.9210	.0036	.0008%

Table 15. Memory and per-epoch time in the original paired runs. VRAM totals retain the source reporting units; times are wall-clock measurements.

		VRAM		Time per epoch	
Backbone	Batch	Base	+WaterVIB	Base	+WaterVIB
HiDDeN	40	24,108	24,220	162 s	165 s
EditGuard	24	24,168	24,715	365 s	372 s
Robust-Wide	2	41,068	40,082	102 s	103 s

J MESSAGE AND HOST INFORMATION IN HIDDEN

J.1 CROSSED MEASUREMENT DESIGN

The study separates output recovery, feature organization, and sensitivity to the readout or estimator. The same frozen HiDDeN pair is used throughout. The measurements ask whether message information reaches the original decoder, how message and host contributions are organized in its features, and which conclusions persist across analysis choices.

Crossed observations. We cross 512 host images with 128 fixed 30-bit messages, giving 65,536 pairs per model and condition. Every host receives every message. We measure the shared 64-dimensional backbone interface and, separately, head features of dimension 30 for the baseline and 128 for the WaterVIB posterior mean. Each model encodes its own watermarked images before attack. Original-decoder results use deterministic inference, including the posterior mean for WaterVIB.

The five base conditions are clean images, Gaussian blur (7×7, $\sigma = 1$), bicubic downsampling from 512^2 to 256^2 and restoration, JPEG quality 70, and SD1.5 VAE reconstruction. The models operate at 256^2 and deliver an 8-bit 512^2 canvas. VAE reconstruction uses posterior-mode encoding and decoding without diffusion denoising. Nested 128×64, 256×64, and 512×64 grids hold the first 64 messages fixed; 512×128 separately expands the message bank.

Generative extension. Global SD1.5, SD2, and ControlNet purification and local SDXL inpainting and RePaint use shared hosts, messages, and masks across models. For each host and attack, the random seed is shared across messages and models. Intersecting valid outputs across conditions and models gives matched cohorts of 59×32 and 112×64. JPEG, VAE, and clean controls use the same retained indices. The cohorts share 58 hosts; they are compared as separate matched cohorts. Attack settings follow Appendix G.

What each measurement describes. Let C and M denote independent host and message indices and F_{cm} the observed feature. The full bank has $H(C) = 9$ and $H(M) = 7$ bits; the expanded AIGC bank has $H(C) = \log_2 112$ and $H(M) = 6$ bits. Index identification and recovery of the 30-bit payload are distinct: near-ceiling message DV can coexist with bit errors. Separately fitted bounds need not preserve the ordering of the unknown true MIs.

Quantity	Interpretation
Decoded-bit MI	Association between each embedded bit and its original-decoder output, averaged over 30 positions (bit/bit).
Neural DV / BA	Fitted variational bounds for finite-bank message or host identification (bits).
Crossed variance	Message, host, and interaction shares of feature variation.
Texture KSG	Dependence on measured host texture descriptors at fixed message (bits).
CKA / Rényi-2	Alignment and information functionals of feature and label kernels.
Explicit-channel MI	Finite-bank information through a specified noisy observation channel (bits).

Sources of variation. Neural summaries use means and sample SD across three fitting seeds, including their seeded partitions. Output intervals use 400 paired host-and-message bootstrap draws. Raw variance-fraction intervals use 300 paired host draws with messages fixed. Kernel SD is across balanced blocks. Nested grids reuse observations and blocks; setting ranges describe sensitivity, while Gaussian Monte Carlo SE describes noise draws on a fixed grid.

Main-text aggregates. For global SD1.5, SD2, and ControlNet, relative DV changes compare the arithmetic means across the three attacks. Mean output MI and variance shares use the same attack set. JPEG classifier-accuracy ranges span the backbone and head interfaces.

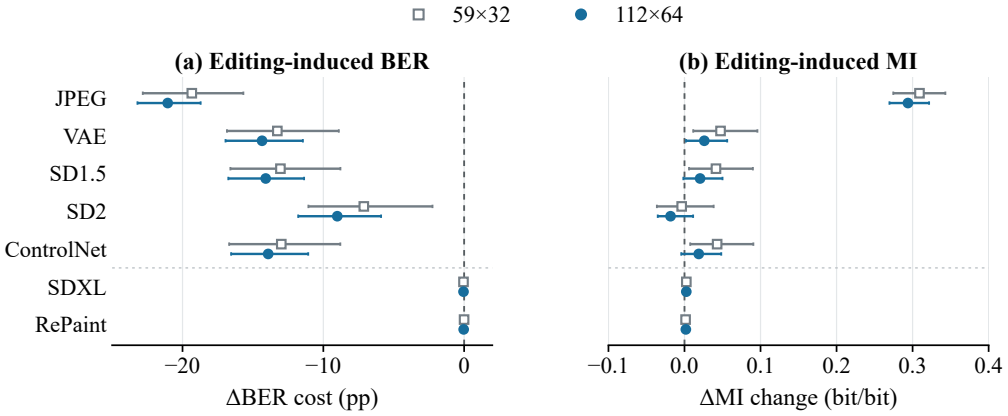
J.2 Output Information and Editing Effects

For each bit, $\widehat{I}(B_\ell; \widehat{B}_\ell)$ comes from its empirical 2×2 frequency table. Averaging over 30 positions measures association; BER measures correctness. Both favor WaterVIB (Table 16).

Table 16. Original-decoder recovery. Pairs are baseline / WaterVIB; MI is bit/bit and its gain has a paired two-way 95% interval.

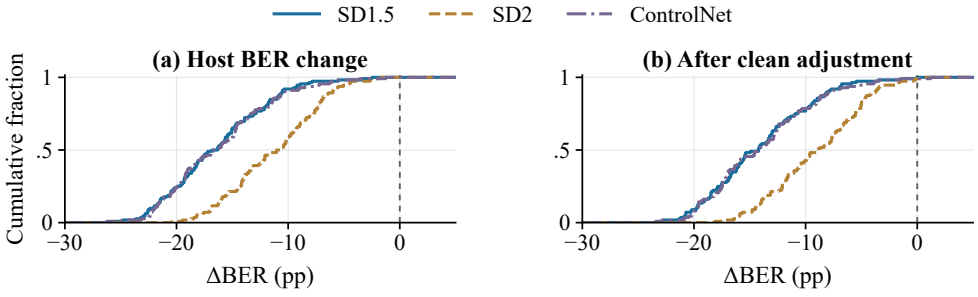
Condition	BER (%)	Bit MI	ΔMI [95% CI]
Base study: 512×128			
Clean	10.67 / 8.75	0.670 / 0.807	0.136 [0.119, 0.149]
Blur	13.86 / 9.01	0.565 / 0.796	0.230 [0.212, 0.243]
Resize	11.77 / 8.87	0.633 / 0.802	0.169 [0.152, 0.181]
JPEG	38.01 / 15.55	0.132 / 0.538	0.406 [0.389, 0.422]
VAE	47.18 / 30.23	0.018 / 0.183	0.165 [0.155, 0.175]
Matched AIGC: 112×64			
Clean	10.58 / 8.32	0.663 / 0.803	0.140 [0.116, 0.155]
JPEG	37.87 / 14.56	0.133 / 0.567	0.434 [0.409, 0.457]
VAE	47.42 / 30.80	0.018 / 0.185	0.166 [0.150, 0.184]
SD1.5	47.50 / 31.15	0.018 / 0.179	0.161 [0.145, 0.178]
SD2	45.18 / 33.91	0.046 / 0.167	0.122 [0.106, 0.141]
ControlNet	47.51 / 31.33	0.019 / 0.178	0.159 [0.143, 0.178]
SDXL	10.66 / 8.35	0.660 / 0.803	0.143 [0.117, 0.157]
RePaint	10.66 / 8.36	0.661 / 0.803	0.142 [0.117, 0.157]

Figure 9. Clean-adjusted editing effects. WaterVIB-minus-baseline mean changes after subtracting each model's clean result, with paired 95% intervals. Negative BER and positive MI changes favor WaterVIB.



Global BER gains remain after clean adjustment, whereas global MI intervals cross zero in the expanded cohort. Local editors add less than 0.08 percentage points of BER to either model in that cohort.

Figure 10. Global BER gains across hosts. CDFs of paired hostwise changes before and after clean adjustment; 112 hosts, 64 messages. Negative values favor WaterVIB.



Coverage across messages and bit positions. The global-editing BER gain covers every measured host and most messages. The bitwise heatmap exposes a less uniform pattern: positive average MI gains coexist with individual positions that lose substantial information. Natural bit order and a shared color scale retain these differences across both cohorts.

Figure 11. All payload positions. WaterVIB-minus-baseline decoded-bit MI for every position. The upper block uses the 512×128 base grid; the lower block uses the 112×64 matched cohort. Blue denotes gain and red denotes loss. The two blocks have different message banks.

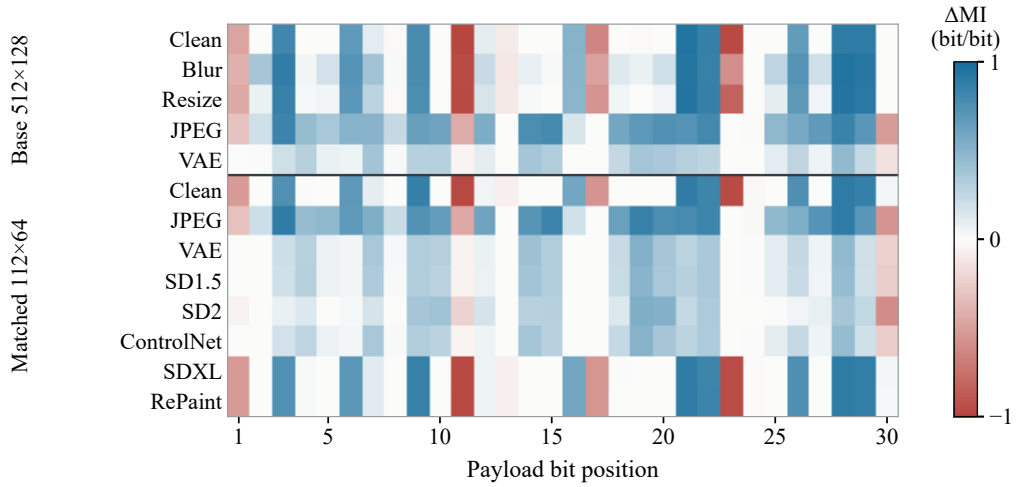


Table 17. Coverage on the matched 112×64 grid. Counts describe lower BER or higher bit MI; adjusted hosts remove each model's clean BER.

Condition	Host BER	Message BER	Bit MI	Adjusted hosts
Clean	112/112	41/64	14/30	0/112
JPEG	112/112	64/64	24/30	112/112
VAE	112/112	61/64	26/30	112/112
SD1.5	112/112	61/64	26/30	112/112
SD2	112/112	56/64	25/30	111/112
ControlNet	112/112	61/64	25/30	111/112
SDXL	112/112	41/64	16/30	59/112
RePaint	112/112	40/64	16/30	54/112

The adjusted clean difference is zero by construction.

Table 18. Pooled and host-conditioned bit MI on the matched grid (bit/bit). Pairs are baseline / WaterVIB; the last columns summarize hostwise gains.

Condition	Pooled MI	Conditional MI	Median Δ	Δ range
Clean	0.663 / 0.803	0.667 / 0.805	0.137	[0.125, 0.176]
JPEG	0.133 / 0.567	0.165 / 0.597	0.437	[0.251, 0.597]
VAE	0.018 / 0.185	0.034 / 0.241	0.212	[0.039, 0.376]
SD1.5	0.018 / 0.179	0.034 / 0.236	0.206	[0.034, 0.356]
SD2	0.046 / 0.167	0.068 / 0.209	0.134	[0.018, 0.287]
ControlNet	0.019 / 0.178	0.035 / 0.235	0.203	[0.018, 0.357]
SDXL	0.660 / 0.803	0.665 / 0.805	0.138	[0.121, 0.184]
RePaint	0.661 / 0.803	0.665 / 0.805	0.138	[0.123, 0.186]

Conditioning the output measurement on host identity also gives positive gains across hosts. Thus, the pooled improvement accompanies stronger within-host bit association. The counts summarize coverage; they are not separate significance tests for each host, message, or bit.

J.3 Sources of Feature Variation

The crossed design decomposes each feature into message, host, and interaction components,

$$F_{cm} = \bar{F} + a_m + b_c + e_{cm}, \qquad a_m = \bar{F}_{\cdot m} - \bar{F}, \quad b_c = \bar{F}_{c\cdot} - \bar{F}. \tag{39}$$

Orthogonality gives $V_{\text{total}} = V_M + V_C + V_{\text{int}}$, with $V_M = \mathbb{E}_m \|a_m\|^2$ and analogous terms. We compare fractions and absolute energies.

Figure 12. Changes in variance allocation. WaterVIB-minus-baseline fractions on the full base grid. Solid points and lines show raw-feature changes and paired 95% host-bootstrap intervals. Hollow points show clean-standardized changes. Zero marks unchanged allocation; units are percentage points.

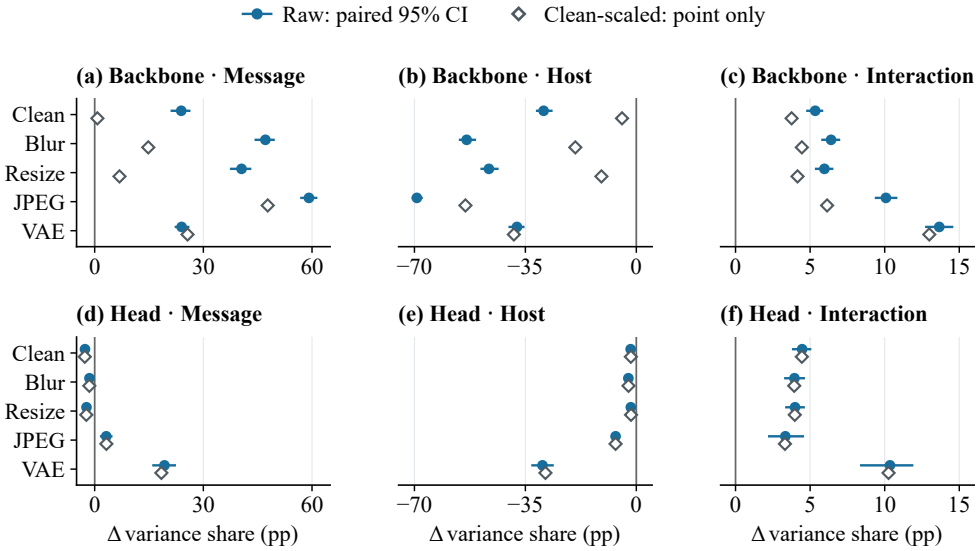


Table 19. Absolute variance energies under full-grid VAE reconstruction. Each cell places baseline above WaterVIB. Scaling uses each model's clean calibration; head widths are 30 and 128.

Interface	Coordinates	Total	Message	Host	Interaction
		1.28×10^{-3}	2.65×10^{-5}	1.23×10^{-3}	2.12×10^{-5}
Backbone	Raw	1.92×10^{-3}	5.01×10^{-4}	1.13×10^{-3}	2.94×10^{-4}
		8.21×10^{0}	1.21×10^{-1}	7.94×10^{0}	1.48×10^{-1}
Backbone	Clean-scaled	8.73×10^{-2}	2.37×10^{-2}	5.07×10^{-2}	1.29×10^{-2}
		7.81×10^{-4}	2.71×10^{-4}	3.97×10^{-4}	1.14×10^{-4}
Head	Raw	7.66×10^{-2}	4.13×10^{-2}	1.62×10^{-2}	1.91×10^{-2}
		4.55×10^{-1}	1.58×10^{-1}	2.30×10^{-1}	6.60×10^{-2}
Head	Clean-scaled	8.67×10^{-2}	4.61×10^{-2}	1.91×10^{-2}	2.15×10^{-2}

Table 20. Variance allocation on the matched 112×64 grid (%). Pairs are baseline / WaterVIB.

Condition	Message	Host	Interaction
Clean	63.55 / 85.75	34.57 / 7.32	1.88 / 6.93
JPEG	8.64 / 66.78	90.00 / 21.95	1.35 / 11.27
VAE	2.29 / 25.71	95.93 / 59.50	1.78 / 14.79
SD1.5	2.24 / 25.20	95.98 / 60.17	1.78 / 14.62
SD2	2.06 / 31.32	96.77 / 52.95	1.17 / 15.73
ControlNet	2.31 / 24.99	95.89 / 60.60	1.79 / 14.41
SDXL	59.41 / 84.47	38.26 / 7.98	2.33 / 7.55
RePaint	60.05 / 84.53	37.61 / 7.92	2.34 / 7.55

JPEG and VAE shift backbone variation toward messages; global purification retains substantial host and interaction components. Energy trends depend on normalization, so reduced host fractions do not imply uniform amplitude compression. Head dimensions also differ.

J.4 IDENTIFICATION AND CLEAN-READOUT TRANSFER

Condition-specific critics estimate M, C, and $C \mid M$. DV is $\mathbb{E}_{\text{joint}}[t] - \log \mathbb{E}_{\text{product}}[e^t]$, with the categorical marginal enumerated and values converted to bits. BA uses $H(Y) + \mathbb{E} \log q(Y \mid Z)$ from the same fitted scores. The DV principle underlies MINE (Belghazi et al., 2018); fitted bounds depend on the critic and optimization (Poole et al., 2019). Independence gives $I((Z, M); C) = I(Z; C \mid M)$.

Critics use two 128-unit ReLU layers, scores $8 \tanh(t/8)$, and AdamW (learning rate 10^{-3}, weight decay 10^{-3}, batch 512, 3,200 updates). Seeds 1701–1703 define 60/20/20% train/validation/test splits. Message tasks split hosts; host tasks split messages. Training-partition statistics standardize features. Validation DV selects a step every 40 updates; conditional critics also receive message bits.

Table 21. Full-grid condition-specific DV (bits), mean ± three-seed SD. B: baseline; W: WaterVIB.

Interface	Condition	M: B	M: W	C: B	C: W	$C \mid M$: B	$C \mid M$: W
Backbone	Clean	$6.999 \pm .000$	$6.996 \pm .001$	$3.409 \pm .532$	$.738 \pm .025$	$3.132 \pm .526$	$.857 \pm .031$
Backbone	Blur	$6.997 \pm .002$	$6.995 \pm .001$	$4.504 \pm .290$	$.783 \pm .040$	$4.828 \pm .099$	$.923 \pm .074$
Backbone	Resize	$6.998 \pm .001$	$6.996 \pm .001$	$4.026 \pm .473$	$.718 \pm .027$	$4.079 \pm .116$	$.853 \pm .060$
Backbone	JPEG	$6.614 \pm .117$	$6.965 \pm .006$	$8.669 \pm .015$	$2.268 \pm .204$	$8.598 \pm .021$	$2.205 \pm .071$
Backbone	VAE	$1.898 \pm .163$	$5.799 \pm .164$	$8.931 \pm .005$	$7.548 \pm .241$	$8.927 \pm .012$	$6.918 \pm .305$
Head	Clean	$6.999 \pm .000$	$6.991 \pm .002$	$.272 \pm .055$	$.958 \pm .024$	$.282 \pm .061$	$.995 \pm .138$
Head	Blur	$6.998 \pm .000$	$6.986 \pm .002$	$.464 \pm .074$	$.916 \pm .155$	$.354 \pm .091$	$1.033 \pm .094$
Head	Resize	$6.999 \pm .000$	$6.989 \pm .002$	$.268 \pm .087$	$.891 \pm .136$	$.294 \pm .050$	$.996 \pm .118$
Head	JPEG	$6.450 \pm .144$	$6.963 \pm .005$	$5.570 \pm .170$	$1.457 \pm .092$	$6.118 \pm .201$	$1.634 \pm .026$
Head	VAE	$1.876 \pm .202$	$5.757 \pm .159$	$8.482 \pm .064$	$5.376 \pm .302$	$8.474 \pm .080$	$5.467 \pm .278$

Table 22. Matched-cohort DV (bits), mean ± three-seed SD. Host identification is conditioned on the message.

Condition	M: Base	M: WaterVIB	$C \mid M$: Base	$C \mid M$: WaterVIB
Clean	6.00 ± 0.00	6.00 ± 0.00	1.70 ± 0.08	0.52 ± 0.43
JPEG	5.17 ± 0.24	5.97 ± 0.02	5.95 ± 0.06	1.45 ± 0.17
VAE	1.06 ± 0.28	4.77 ± 0.11	6.63 ± 0.02	4.65 ± 0.24
SD1.5	1.01 ± 0.17	4.70 ± 0.11	6.61 ± 0.03	4.69 ± 0.20
SD2	1.91 ± 0.50	5.21 ± 0.09	6.47 ± 0.11	3.50 ± 0.09
ControlNet	0.94 ± 0.39	4.77 ± 0.13	6.61 ± 0.04	4.73 ± 0.20
SDXL	6.00 ± 0.00	6.00 ± 0.00	1.95 ± 0.14	0.49 ± 0.44
RePaint	6.00 ± 0.00	6.00 ± 0.00	1.80 ± 0.19	0.48 ± 0.44

Table 23. Clean-trained classifiers transferred without refitting on the full grid. Accuracy (%), mean ± three-seed SD; M and C identify message and host indices.

Interface	Condition	M: Base	M: WaterVIB	C: Base	C: WaterVIB
Backbone	Clean	100.00 ± 0.00	100.00 ± 0.00	21.25 ± 1.84	1.27 ± 0.21
Backbone	Blur	98.54 ± 1.48	100.00 ± 0.00	1.35 ± 0.24	0.48 ± 0.08
Backbone	Resize	99.97 ± 0.04	100.00 ± 0.00	2.85 ± 0.33	0.77 ± 0.13
Backbone	JPEG	13.30 ± 6.46	91.56 ± 1.29	0.41 ± 0.16	0.20 ± 0.00
Backbone	VAE	1.45 ± 0.46	5.95 ± 2.87	0.04 ± 0.05	0.20 ± 0.00
Head	Clean	100.00 ± 0.00	100.00 ± 0.00	0.74 ± 0.09	1.00 ± 0.16
Head	Blur	99.02 ± 1.06	100.00 ± 0.00	0.21 ± 0.02	0.32 ± 0.03
Head	Resize	99.99 ± 0.00	100.00 ± 0.00	0.21 ± 0.03	0.59 ± 0.04
Head	JPEG	12.89 ± 5.21	94.60 ± 1.47	0.20 ± 0.00	0.20 ± 0.00
Head	VAE	1.27 ± 0.05	8.67 ± 2.26	0.19 ± 0.00	0.19 ± 0.00

The clean-trained classifier uses one 128-unit ReLU layer, cross-entropy, and AdamW (learning rate 0.002, weight decay 0.001, batch 256, 3,200 updates). It uses the same seeded partitions and selects a step by clean validation every 20 updates.

WaterVIB's clean classifier transfers well under JPEG but poorly under VAE, while separately fitted critics still identify VAE message structure. These readouts distinguish condition-specific identifiability from reuse of a clean rule.

J.5 Conditional Texture Dependence

For each fixed message, KSG (Kraskov et al., 2004) estimates dependence between four feature PCs and two PCs of a seven-dimensional host descriptor. The descriptor includes RGB means, grayscale variation, gradient, Laplacian, and high-pass statistics. We average the estimates over messages.

Feature standardization and PCA use a fixed clean block of 76 hosts and 64 messages. Texture coordinates and reference scales use the full base grid and are reused for nested subsets. Global scaling uses each model/interface/condition's reference grid; within-message scaling normalizes each message group. We retain $k = 3, 5, 10$ and both scalings. Shuffled controls permute host-descriptor pairing at fixed message.

Figure 13. Neighbor-count and scaling sensitivity. Full-grid WaterVIB-minus-baseline texture KSG. Each curve follows one condition as k changes; rows distinguish interfaces and columns distinguish scaling rules. Negative values indicate lower measured texture dependence.

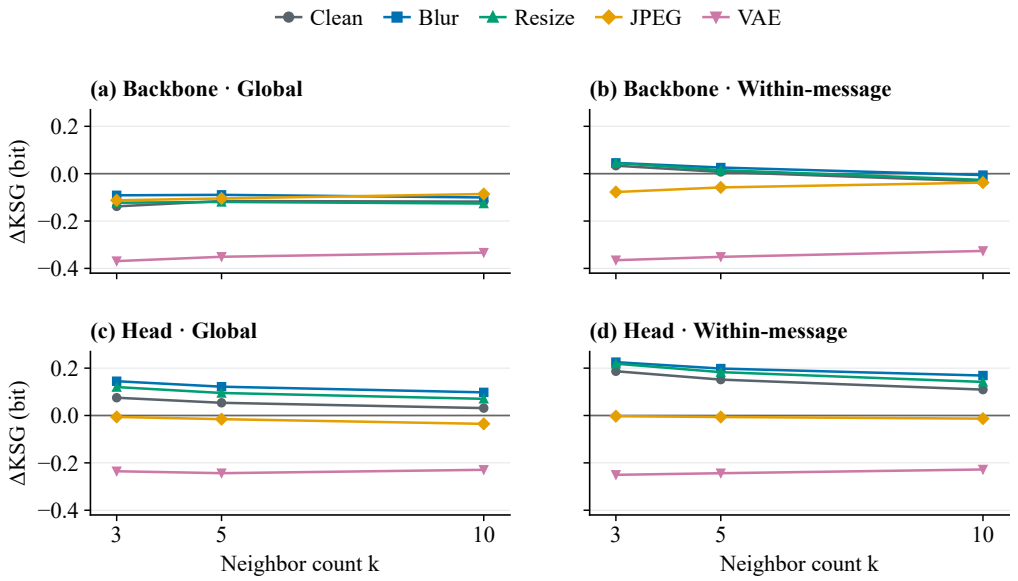


Table 24. Texture KSG (bits), baseline / WaterVIB at full-grid $k = 5$. BB: backbone. Counts and ranges span four grids and six settings; null ranges include both models.

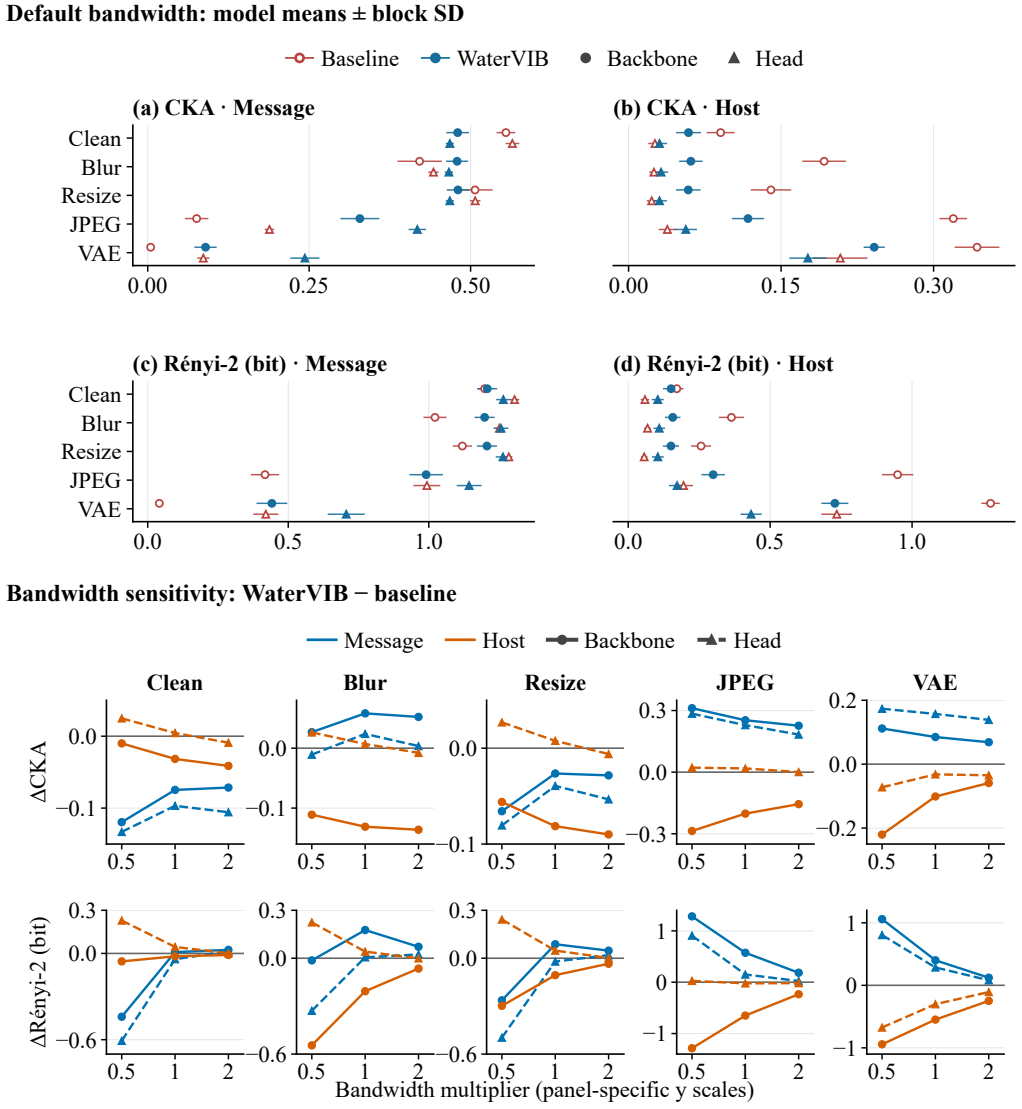
Interface / condition	Global	Within M	$\Delta < 0$	Δ range	Null range
BB / Clean	.3187 / .2031	.3581 / .3654	13/24	$[-.1538, .1783]$	$[-.0590, .0758]$
BB / Blur	.3594 / .2700	.3905 / .4164	11/24	$[-.1513, .2165]$	$[-.0513, .0809]$
BB / Resize	.3598 / .2405	.3827 / .3974	11/24	$[-.1561, .1877]$	$[-.0607, .0718]$
BB / JPEG	.4473 / .3425	.4302 / .3721	18/24	$[-.1815, .1092]$	$[-.0124, .0660]$
BB / VAE	.6903 / .3394	.6796 / .3284	24/24	$[-.4119, -.0565]$	$[-.0349, .0547]$
Head / Clean	.1339 / .1875	.1917 / .3433	1/24	$[-.0018, .2862]$	$[-.0176, .0606]$
Head / Blur	.1378 / .2595	.2036 / .4018	0/24	$[.0621, .3354]$	$[-.0341, .0588]$
Head / Resize	.1324 / .2275	.1950 / .3781	0/24	$[.0366, .3176]$	$[-.0259, .0637]$
Head / JPEG	.3384 / .3230	.3934 / .3870	12/24	$[-.0908, .1565]$	$[-.0229, .0452]$
Head / VAE	.5205 / .2770	.5108 / .2671	24/24	$[-.2505, -.0007]$	$[-.0267, .0358]$

VAE has lower dependence throughout all 48 grid–interface–setting combinations. The mild-condition head estimates often increase. JPEG is sensitive to which hosts and messages are included: the backbone changes direction after host expansion, and the head's within-message estimates change direction when the message bank doubles. Reconstruction therefore provides the most consistent reduction; it is not a uniform response across attacks.

J.6 Message and Host Label Geometry

Balanced 32×64 blocks cover each grid (32 blocks in full). RBF kernels use complete, clean-standardized interface vectors and 0.5, 1, or 2 times each block/model/condition's median-distance bandwidth. Label kernels indicate index equality. Centered kernel alignment (CKA) is the normalized inner product of centered feature and label kernels (Kornblith et al., 2019). The matrix Rényi-2 functional is $S_2(K_Z)+S_2(K_Y)-S_2(K_Z \circ K_Y)$, with $S_2(K) = -\log_2 \mathrm{tr}[(K/\mathrm{tr}\,K)^2]$ (Sanchez Giraldo et al., 2015). It measures kernel geometry and is distinct from Shannon MI.

Figure 14. Label geometry and bandwidth sensitivity. Upper: full-grid means and block SD at bandwidth factor 1. Lower: WaterVIB-minus-baseline bandwidth sweeps for both targets and interfaces; scales vary by condition.



Under VAE, both kernel quantities consistently raise message alignment and lower host alignment across interfaces, bandwidths, and grids. JPEG head host CKA can instead rise while its Rényi functional falls. Both metrics share kernels and blocks. The default VAE backbone baseline message functional (0.041) is below its shuffled value (0.069). Nulls permute target labels within each 2,048-pair block, holding feature kernels fixed; no null subtraction is applied.

J.7 Sample Inclusion and Disjoint-Bank Replication

The fixed-64-message sequence tests incremental host inclusion with unchanged feature transforms. The final point separately doubles the message bank. Output-MI gain intervals stay positive in every base condition and grid; feature-level quantities show more sensitivity, including the JPEG reversals.

Figure 15. Nested base-study stability. Curves keep 64 messages fixed; the separate Full point uses 512×128. Output-MI intervals use paired two-way bootstrap and raw-variance intervals use host bootstrap. Texture KSG uses global scaling and $k = 5$. Nested observations are reused.

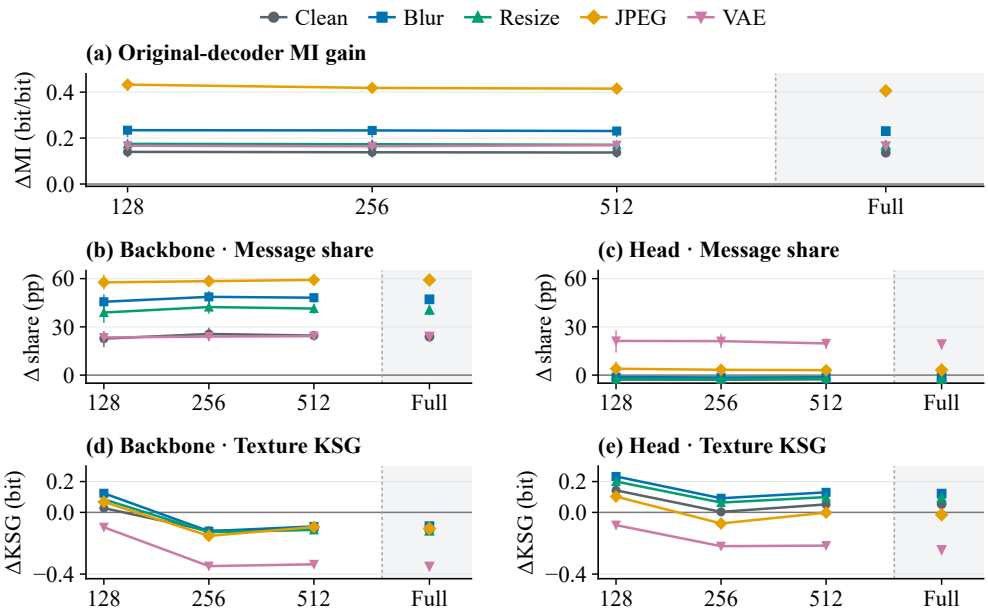


Table 25. Smaller matched AIGC cohort (59×32). Pairs are baseline / WaterVIB; decoded MI is bit/bit and DV is bits. The retained host cohorts are not strictly nested.

Condition	BER (%)	Bit MI	ΔMI 95% CI	DV M	DV $C \mid M$
Clean	10.55 / 7.68	0.666 / 0.799	[0.096, 0.159]	4.991 / 4.922	1.215 / 0.283
JPEG	36.65 / 14.44	0.133 / 0.575	[0.403, 0.469]	4.305 / 4.893	4.794 / 1.112
VAE	46.27 / 30.14	0.019 / 0.200	[0.154, 0.204]	0.887 / 4.145	5.628 / 3.367
SD1.5	46.44 / 30.53	0.020 / 0.194	[0.149, 0.198]	0.793 / 4.104	5.662 / 3.421
SD2	44.44 / 34.44	0.044 / 0.174	[0.106, 0.151]	1.751 / 4.202	5.473 / 2.536
ControlNet	46.40 / 30.55	0.020 / 0.196	[0.150, 0.199]	0.895 / 4.071	5.653 / 3.441
SDXL	10.64 / 7.72	0.663 / 0.798	[0.096, 0.161]	4.985 / 4.994	1.299 / 0.312
RePaint	10.61 / 7.74	0.664 / 0.798	[0.095, 0.160]	4.984 / 4.995	1.274 / 0.327

Table 26. Disjoint 64×32 bank, unchanged frozen checkpoints. Pairs are baseline / WaterVIB; MI-gain intervals use matched saved bootstrap draws.

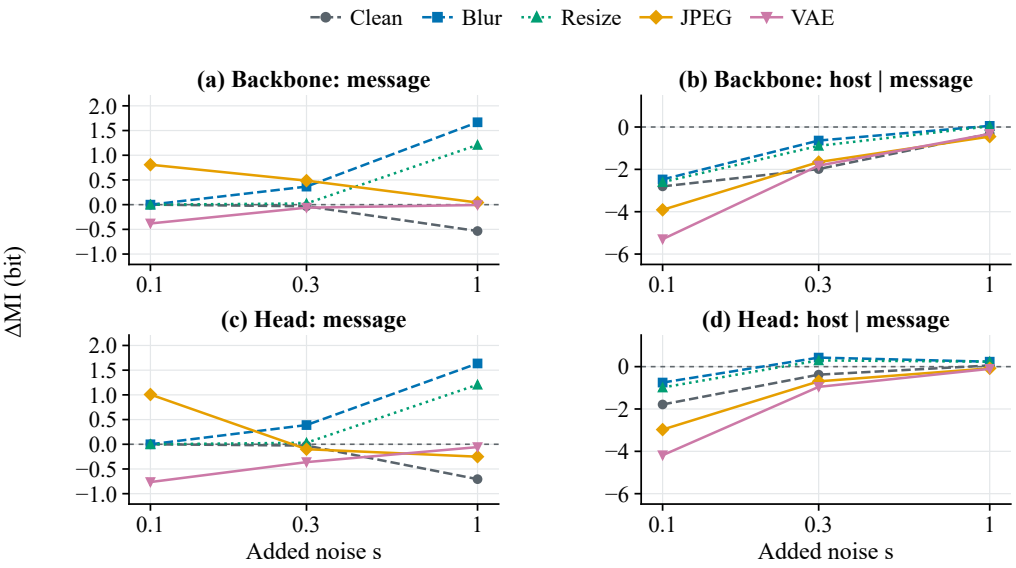
Condition	BER (%)	Bit MI (bit/bit)	ΔMI 95% CI
Clean	9.10 / 8.23	0.696 / 0.791	[0.052, 0.120]
Blur	12.30 / 8.42	0.582 / 0.782	[0.147, 0.224]
Resize	10.27 / 8.30	0.652 / 0.787	[0.088, 0.160]
JPEG	38.33 / 14.59	0.119 / 0.546	[0.386, 0.458]
VAE	47.45 / 28.40	0.019 / 0.206	[0.161, 0.214]

Filtering changes AIGC host membership as well as size (Figure 9). The disjoint 64×32 bank has no image-hash or message-vector overlap with the full bank and uses the same frozen checkpoints. All five output-MI gain intervals remain positive.

J.8 Explicit Measurement Channels

The added-noise diagnostic projects clean-standardized features to 16 PCs, normalizes by clean RMS, and evaluates $Y_s = z + s\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ and $s \in \{0.1, 0.3, 1\}$. Transforms stay fixed across conditions. Gaussian-mixture densities give finite-bank index information, with expectations evaluated using two noise draws per pair. Monte Carlo SE describes noise-sampling error on this fixed grid.

Figure 16. Information through an added-noise channel. Full-grid WaterVIB-minus-baseline message and conditional host information as noise scale changes. Conditions and interfaces are shown separately. Noise levels define different observation channels.



Under VAE reconstruction, both message and conditional host information decrease throughout the noise sweep and all four grids. This channel gives a different view from deterministic payload decoding, and shows why a representation-level measurement must identify its observation rule.

J.9 Calibration and Estimator Sensitivity

Known-answer checks give 0.5373 bit against 0.5310 for a binary channel and KSG 0.4974 against 0.4857 for a correlated Gaussian. Ross and its reference implementation both give 1.00014 on a separate, well-separated binary-class control.

Table 27. Full-grid Ross estimates with eight PCs and $k = 5$ (bits), baseline / WaterVIB. Full-grid shuffled ranges: M $[-0.0031, 0.0050]$, C $[-0.0024, 0.0043]$.

Condition	M: Backbone	M: Head	C: Backbone	C: Head
Clean	6.975 / 6.984	6.982 / 6.997	0.685 / 0.280	0.003 / − 0.203
Blur	6.943 / 6.984	6.928 / 6.997	1.127 / 0.306	0.149 / − 0.184
Resize	6.972 / 6.985	6.953 / 6.998	0.903 / 0.277	0.029 / − 0.193
JPEG	4.771 / 6.930	5.113 / 6.968	3.829 / 0.936	2.065 / − 0.010
VAE	0.353 / 4.670	1.183 / 5.358	8.060 / 2.966	6.457 / 1.409

Table 28. Host-Ross controls (bits). Synthetic features contain no host signal. Real rows use clean features; within-message and global shuffles answer different questions.

Control	Truth	Observed	Global shuffle	Within-M shuffle
Synthetic $\sigma = 0.01$	0	-0.2778	0.0021	-0.2777
Synthetic $\sigma = 0.10$	0	-0.2766	0.0018	-0.2774
Synthetic $\sigma = 1.00$	0	-0.0711	0.0001	-0.0699
Base / Backbone	–	0.6846	-0.0019	-0.2052
Base / Head	–	0.0032	0.0035	-0.2576
WaterVIB / Backbone	–	0.2803	0.0001	-0.2269
WaterVIB / Head	–	-0.2026	0.0018	-0.2683

Crossed zero-host-information controls yield negative finite-sample Ross estimates (Ross, 2014). Global shuffling approaches zero; within-message shuffling preserves the bias. Disjoint-bank transforms use 38 clean hosts and 32 messages. There, JPEG/VAE retain higher message and lower host estimates at both interfaces; clean/resize message estimates are slightly lower, and some head-host differences reverse.

Table 29. Neural fitting diagnostics across four grids and two interfaces. Ranges span condition means (bits); the null shuffles labels for a fixed fitted critic.

Target	Model	Step ≥ 3000	Validation–test range	Null DV range
M	Base	83/120	$[-0.154, 0.236]$	$[-15.700, -2.609]$
M	WaterVIB	93/120	$[-0.067, 0.084]$	$[-14.388, -5.824]$
C	Base	33/120	$[-0.285, 0.698]$	$[-14.723, -0.520]$
C	WaterVIB	3/120	$[-0.220, 0.424]$	$[-10.995, -1.693]$
$C \mid M$	Base	19/120	$[-0.301, 0.497]$	$[-13.597, -0.543]$
$C \mid M$	WaterVIB	1/120	$[-0.285, 0.456]$	$[-10.051, -1.796]$

Fixed-critic nulls shuffle test labels without retraining; the table also exposes late-selected fits and validation–test gaps. BA and DV share fitted scores. On earlier 128×64 controls, extending 800 to 3,200 updates changes message DV by -0.004 to $+0.023$ bit; the largest shift is $+0.195$ for baseline VAE head host DV.

Table 30. Earlier backbone Ross sensitivity over 4/8/16 PCs and $k = 3/5/10$, using its original calibration. Ranges span nine settings per condition (bits).

Condition	Δmessage range	Δhost range	Shuffled range
Clean	$[0.0195, 0.7048]$	$[-0.4945, -0.2552]$	$[-0.0167, 0.0181]$
Blur	$[-0.0100, 0.7927]$	$[-1.0112, -0.7412]$	$[-0.0131, 0.0166]$
Resize	$[-0.0212, 0.6856]$	$[-0.7347, -0.4795]$	$[-0.0137, 0.0181]$
JPEG	$[1.0467, 2.8987]$	$[-3.0384, -2.2938]$	$[-0.0081, 0.0225]$
VAE	$[2.1247, 4.3413]$	$[-4.1914, -3.5914]$	$[-0.0183, 0.0116]$