FROM FIXED JND TO LEARNED ATTENUATION IN INVISIBLE IMAGE WATERMARKING

Anonymous authors
Paper under double-blind review

ABSTRACT

In invisible image watermarking, the attenuation map sets the per-pixel strength of the watermark residual. We ask whether replacing fixed just-noticeable-difference (JND) attenuation with a learned attenuation map improves the quality–robustness trade-off, and whether any robustness gain survives when the embedded distortion, rather than the embedding coefficient, is held fixed. Freezing Pixel Seal's embedder and extractor, we train a U-Net that predicts the attenuation map. At the same coefficient, the base learned map raises bit accuracy under a combined JPEG–crop–brightness attack by 3.81 percentage points (pp) but lowers PSNR by 5.37 dB; adding an LPIPS loss and Shape-Aligned Attenuation (SAA), which aligns the mean-normalized learned and JND maps, recovers 2.05 dB at a 0.31 pp accuracy cost. At matched PSNR, however, this advantage disappears: fixed JND takes a small but significant lead (0.36–0.38 pp over the best learned condition at all but the highest target), and among the learned conditions the robustness differences come from SAA ($+0.55$–2.59 pp, with or without the LPIPS loss). The learned conditions' lower LPIPS at matched PSNR also largely vanishes at matched robustness, where fixed JND needs the least distortion. In exploratory tests on MaskWM, WAM, and VideoSeal, matching PSNR likewise narrows the JPEG-attack gap to under 1.2 pp (learned attenuation stays slightly ahead on VideoSeal). With Pixel Seal's backbone frozen, learned attenuation therefore improves on fixed JND neither in robustness at equal distortion nor clearly in quality at equal robustness, and single-coefficient comparisons overstate its benefit.

1 INTRODUCTION

Invisible image watermarking aims to embed a message while preserving an image's visual quality and to recover that message after transformations such as compression or geometric distortion (Zhu et al., 2018; Tancik et al., 2020; Souček et al., 2025), so a system must trade imperceptibility against robustness (Zhu et al., 2018; Sander et al., 2025; Souček et al., 2025). Where it sits on that trade-off depends not only on the watermark's overall embedding strength but also on how the signal is distributed across the image (Souček et al., 2025; Zhang et al., 2008). Pixel Seal (Souček et al., 2025) controls that distribution with a just-noticeable-difference (JND) map (Zhang et al., 2008), which estimates, at each location, how much change is tolerable based on luminance adaptation and contrast masking; Pixel Seal applies this map to the embedder's residual as a per-pixel attenuation. Unlike the embedder and extractor, this attenuation follows a predefined formula and is never optimized by the watermarking objective (Souček et al., 2025), so it remains unclear how the quality–robustness relationship changes when attenuation itself is learned, and whether any robustness gain would survive a comparison at equal distortion rather than at an equal embedding coefficient.

This work makes that attenuation learnable and asks what it changes. We freeze the pretrained embedder and extractor of Pixel Seal and train a U-Net-based (Ronneberger et al., 2015) module that predicts a per-pixel attenuation map from the input image. This setup isolates the effect of learning attenuation from any change to the backbone itself. The base learned attenuation raises post-attack bit accuracy but substantially lowers image quality; to mitigate that loss, we add an LPIPS-based perceptual loss and Shape-Aligned Attenuation (SAA), which mean-normalizes the learned and JND maps before aligning their relative spatial distributions, and so uses the JND map's spatial guidance without forcing the learned map's absolute magnitude to match it.

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.

On SA-1B at a fixed global embedding strength ($\alpha = 0.2$), combining both regularizers (LPIPS+SAA) raises PSNR by 2.05 dB over the base learned attenuation, and adding SAA to the base or to the LPIPS condition contributes $+0.97$ and $+1.02$ dB, respectively, while bit accuracy under the Combined attack (JPEG, crop, and brightness in sequence) changes by at most 0.31 percentage points (pp) in either direction (Table 1). We report these differences as observations from a limited set of three training seeds.

Because a fixed α implies a different actual distortion level for each condition, we also compare each condition at matched mean PSNR by independently calibrating its α. This comparison changes the conclusion: NATIVE leads every learned condition at all but the highest target, by a small but significant 0.36–0.38 pp over the best of them; among the learned conditions, the two SAA variants are highest; and the learned conditions show lower LPIPS and higher SSIM at matched PSNR (Figure 2), but the LPIPS advantage largely disappears at matched robustness. On MaskWM, WAM, and VideoSeal (exploratory), the fixed-coefficient change varies in direction by backbone, but matching PSNR likewise narrows the JPEG-attack gap, with learned attenuation slightly ahead only on VideoSeal.

Our main contributions are as follows.

- We isolate the attenuation component of a pretrained watermarking system – freezing the embedder and extractor – to study how replacing fixed-JND attenuation with a learned predictor changes the quality–robustness trade-off.
- We show that the robustness gap between fixed-JND and learned attenuation at a fixed coefficient largely disappears once the distortion budget is matched – on Pixel Seal and under extreme downscaling (resize-only), and on JPEG attacks for MaskWM, WAM, and VideoSeal – indicating that fixed-coefficient comparisons of this kind can be misleading.
- We find that unconstrained learned attenuation spreads its response more broadly than fixed JND, and introduce Shape-Aligned Attenuation (SAA) to recover a JND-like allocation without constraining global strength, separating its contribution from that of the LPIPS loss via a 2×2 ablation and a matched-PSNR comparison, which shows that SAA is what drives the learned conditions' robustness at matched PSNR.

2 RELATED WORK

JND-based neural watermarking. Several neural watermarking methods use JND-based modulation to improve imperceptibility, scaling the watermark signal's local strength according to the image's perceptual characteristics. WAM (Sander et al., 2025), VideoSeal (Fernandez et al., 2024), and Pixel Seal (Souček et al., 2025) all use JND-based perceptual modulation during embedding, and MaskWM (Hu et al., 2025) likewise uses a JND module to modulate the watermark signal's per-location magnitude. In these methods, however, the JND map is fixed and not trained with the watermarking objective. We focus on this fixed-JND attenuation and replace its per-pixel role with a learnable module, primarily in Pixel Seal, with exploratory tests on the other three.

Learned JND and JND-guided supervision. Prior work has also studied learning JND itself from data: RGB-JND (Jin et al., 2022) and MetaJND (Wang et al., 2024) estimate an image's perceptual sensitivity with neural networks, and Qin et al. (2026) use JND as a training signal to shape the watermark residual's spatial distribution. Our SAA likewise uses JND as a training signal but differs along three axes: its *target* is a separately learned attenuation map $A_\phi(x)$ rather than the residual (§3); its *normalization* divides both maps by their own spatial means first (Eq. 8), making the loss invariant to a positive global rescaling of the learned map and leaving its magnitude to the other losses, unlike a loss on unnormalized maps or the residual; and its *setting* keeps the embedder and extractor frozen, so the JND signal cannot be absorbed by retraining the backbone. Empirically, adding SAA multiplies the attenuation map's high-frequency and gradient statistics by 2.6–$4.2\times$ while changing its mean magnitude by only about 10% (§4.2). A related normalize-then-align strategy (using L2 normalization) is used for feature maps in Attention Transfer (Zagoruyko & Komodakis, 2017).

Adaptive embedding-strength control. Other work adaptively controls embedding strength according to image characteristics. Bagheri et al. (2020) determine appropriate embedding strength

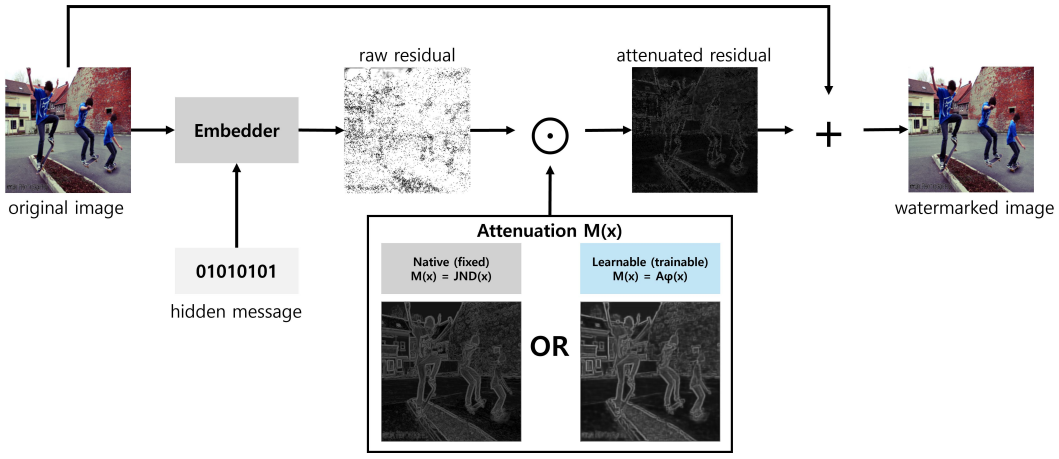


Figure 1: **Overview.** The frozen Pixel Seal embedder turns the input image x and a hidden message into a raw residual w, which the attenuation map $M(x)$ scales pixel-wise before it is added back to x: $x_w = x + \alpha\,(w \odot M(x))$ (Eq. 1). Pixel Seal uses the fixed $M(x) = \mathrm{JND}(x)$ (NATIVE); we replace it with a learned $M(x) = A_\phi(x)$, keeping the embedder and extractor frozen. During training, A_ϕ is the only watermarking network updated; it receives gradients from the message loss through the frozen extractor, the adversarial loss of a jointly trained patch discriminator, LPIPS, and SAA, which aligns the mean-normalized $A_\phi(x)$ with $\mathrm{JND}(x)$. Residuals are magnified.

for local image regions, and Wang et al. (2023) propose Adaptor, which adaptively selects an embedding-strength coefficient from the input image and watermark message. In contrast, this work treats the dense per-pixel attenuation map itself (the role filled by Pixel Seal's fixed-JND map) as the learning target (§3). Because they use different backbones, data, and protocols, we do not compare against them numerically; our goal is to characterize what making fixed JND learnable changes under a controlled, fixed-backbone setting.

3 METHOD

3.1 OVERALL ARCHITECTURE

We keep Pixel Seal's embedder and extractor unchanged and replace only the fixed JND map, which sets the residual's per-pixel embedding strength, with a learnable attenuation map A_ϕ (Figure 1).

Let an input image $x \in [0,1]^{3\times H\times W}$ and an n_{bits}-bit binary message $m \in \{0,1\}^{n_{\text{bits}}}$ be given. We downsample x to the embedder's processing resolution and use the pretrained Pixel Seal embedder Emb, which takes the luminance (Y) channel, to produce a low-resolution single-channel residual and upsample it back to obtain $w \in \mathbb{R}^{1\times H\times W}$. Writing the attenuation map generally as $M(x) \in [0,1]^{1\times H\times W}$, the watermarked image is

$$x_w = x + \alpha\,(w \odot M(x))\,, \tag{1}$$

where α is a global scale coefficient and $\odot$ denotes the pixel-wise product; both w and $M(x)$ are broadcast identically across the three RGB channels, so the watermark changes only luminance. α sets the overall embedding strength, while $M(x)$ sets its relative expression at each image location. Pixel Seal's existing formulation sets $M(x) = \mathrm{JND}(x)$ using a non-learned formula. We instead predict the attenuation map with a learnable module, $M(x) = A_\phi(x)$, so the learned embedding is

$$x_w^{\text{learn}} = x + \alpha\,(w \odot A_\phi(x))\,, \tag{2}$$

where A_ϕ is an all-convolutional U-Net-style network with 116,225 trainable parameters, producing a single-channel attenuation map with values in $(0,1)$ via a sigmoid output. The encoder downsamples the feature map to $1/4$ resolution using two stages of 3×3 strided convolutions; the de-

coder upsamples by a factor of two at each stage using nearest-neighbor interpolation followed by a convolution, and each upsampled map is concatenated with the encoder feature map of matching resolution via a skip connection. We use strided convolutions instead of pooling, following the all-convolutional design of Springenberg et al. (2015), and upsampling-plus-convolution instead of transposed convolution to avoid its checkerboard artifacts (Odena et al., 2016). Our learned attenuation does not alter the watermark residual w produced by the embedder; it only controls how strongly that residual is expressed at each image location.

During training, watermarked images undergo stochastic geometric, valuemetric, and compression transformations $\mathcal{T}$, after which they are downsampled to the extractor's processing resolution to recover the message:

$$\tilde{x}_w = \mathcal{R}_{\downarrow}(\mathcal{T}(x_w)), \qquad \tilde{m} = \mathrm{Ext}(\tilde{x}_w), \tag{3}$$

where Ext is the pretrained Pixel Seal extractor and $\tilde{m} \in [0,1]^{n_{\mathrm{bits}}}$ is the predicted probability for each message bit.

3.2 Training Objective

During training, we freeze the pretrained Pixel Seal embedder Emb and extractor Ext and optimize the newly added attenuation module A_ϕ, together with a patch discriminator D trained adversarially alongside it (Appendix A). Although Ext is frozen, gradients flow through it and through D to their inputs, so the message and discriminator-based losses reach the attenuation module. The total training objective combines four terms:

$$\mathcal{L}_{\mathrm{total}} = \lambda_{\mathrm{msg}}\mathcal{L}_{\mathrm{msg}} + \lambda_{\mathrm{adv}}\mathcal{L}_{\mathrm{adv}} + \lambda_{\mathrm{lpips}}\mathcal{L}_{\mathrm{lpips}} + \lambda_{\mathrm{SAA}}\mathcal{L}_{\mathrm{SAA}}. \tag{4}$$

$\mathcal{L}_{\mathrm{msg}}$ and $\mathcal{L}_{\mathrm{adv}}$ form the base objective $\mathcal{L}_{\mathrm{base}} = \lambda_{\mathrm{msg}}\mathcal{L}_{\mathrm{msg}} + \lambda_{\mathrm{adv}}\mathcal{L}_{\mathrm{adv}}$ shared by every learned condition; $\mathcal{L}_{\mathrm{lpips}}$ and $\mathcal{L}_{\mathrm{SAA}}$ are the two regularization terms this work adds to mitigate the quality loss that $\mathcal{L}_{\mathrm{base}}$ alone induces (§4.2).

Message and discriminator-based losses. The message-recovery loss is the binary cross-entropy between the message probability $\tilde{m}$ extracted from the transformed watermarked image and the original message m:

$$\mathcal{L}_{\mathrm{msg}} = -\frac{1}{n_{\mathrm{bits}}} \sum_{k=1}^{n_{\mathrm{bits}}} \left[m_k \log \tilde{m}_k + (1 - m_k) \log(1 - \tilde{m}_k) \right]. \tag{5}$$

$\mathcal{L}_{\mathrm{msg}}$ shapes the attenuation map's spatial distribution and overall magnitude so that the frozen extractor can recover the original message even after various image transformations. The discriminator-based loss for imperceptibility is defined using the patch discriminator D as

$$\mathcal{L}_{\mathrm{adv}} = -\operatorname{mean}(D(x_w)), \tag{6}$$

where $\operatorname{mean}(\cdot)$ averages the patch-based discriminator's spatial output, encouraging the watermarked image to be difficult for the discriminator to distinguish from the original.

Perceptual distance regularization. Neither loss above directly constrains the reference-based deviation between the original and watermarked images. We therefore add an LPIPS-based regularization term that constrains the perceptual feature distance between the two images:

$$\mathcal{L}_{\mathrm{lpips}} = \mathrm{LPIPS}(x_w, x). \tag{7}$$

$\mathcal{L}_{\mathrm{lpips}}$ guides the attenuation module so that the perceptual feature distance between the original image x and the watermarked image x_w stays small. We treat LPIPS not as a complete proxy for human perception but as an auxiliary regularizer, and isolate its effect with a dedicated condition (§3.2). The loss takes $[0,1]$ inputs; reported LPIPS values use the standard $[-1,1]$ input (Appendix A).

Shape-Aligned Attenuation. The perceptual-distance loss constrains the overall perceptual difference between the original and watermarked images, but it does not directly specify what spatial distribution the learned attenuation map should have within the image. We introduce Shape-Aligned Attenuation (SAA) to leverage the spatial perceptual structure already provided by the existing JND map. Directly aligning $A_\phi(x)$ to $\mathrm{JND}(x)$ would not only constrain the attenuation map's spatial structure but also force it to reproduce the JND map's absolute magnitude; our goal is instead only to encode the spatial relationship between locations where relatively larger changes are tolerable and locations where smaller changes are required. To this end, we normalize the learned attenuation map and the JND map by their own spatial means. Using a small constant $\epsilon > 0$ for numerical stability, we define the mean-normalized attenuation map and JND map as

$$\widetilde{A}(x) = \frac{A_\phi(x)}{\mu_A + \epsilon}, \qquad \widetilde{J}(x) = \frac{\mathrm{JND}(x)}{\mu_J + \epsilon}, \tag{8}$$

where $\mu_A = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\left[A_\phi(x)\right]_{i,j}$ and $\mu_J = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\left[\mathrm{JND}(x)\right]_{i,j}$ are each map's own spatial mean, computed per image rather than over the batch. We then define the SAA loss as the pixel-wise mean squared error between the two normalized maps:

$$\mathcal{L}_{\mathrm{SAA}} = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\left(\left[\widetilde{A}(x)\right]_{i,j} - \left[\widetilde{J}(x)\right]_{i,j}\right)^2. \tag{9}$$

Mean normalization separates each map's overall magnitude from its relative spatial structure: ignoring ϵ, for any constant $c > 0$ multiplying the entire attenuation map, $cA_\phi(x)/\operatorname{mean}(cA_\phi(x)) = A_\phi(x)/\operatorname{mean}(A_\phi(x))$ holds, so the normalized map is invariant to positive global scaling; in practice this property holds approximately in the typical regime where $\mu_A \gg \epsilon$. Consequently, SAA does not force the attenuation map's absolute magnitude to match the JND map; instead, it guides the spatial pattern of relatively larger and smaller attenuation values to align with JND, while the map's overall magnitude is left to the other losses.

Comparison conditions. To separate the effects of learned attenuation and each regularization term, we compare five conditions. NATIVE is the released Pixel Seal checkpoint with its original fixed-JND attenuation and no additional training in this work (this does not imply that perceptual or adversarial losses were absent from the checkpoint's original pretraining). The four learned conditions all start from the NATIVE checkpoint, with the embedder and extractor frozen and the attenuation module trained on a different subset of $\mathcal{L}_{\mathrm{total}}$ (Eq. 4): FREE is trained with only $\mathcal{L}_{\mathrm{msg}}$ and $\mathcal{L}_{\mathrm{adv}}$; LPIPS and SAA each add one regularization term to the base objective; LPIPS+SAA adds both. Comparing NATIVE and FREE isolates the effect of replacing fixed JND with learned attenuation on the same pretrained backbone; comparisons among the learned conditions isolate the individual and joint effects of the LPIPS and SAA regularizers.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Backbone and training. We use the released Pixel Seal (Souček et al., 2025) checkpoint (256-bit messages) as the base backbone. In learned conditions, we freeze the pretrained embedder and extractor and train the U-Net-based attenuation module (with the discriminator) for 10 epochs of 1,000 iterations each (10,000 iterations total) on COCO (Lin et al., 2014), with the same batch size, schedule, and optimizer for every learned condition; reported metrics use the final (10th) epoch checkpoint. Training and the fixed-coefficient evaluation both use $\alpha = 0.2$. All learned conditions use $\lambda_{\mathrm{msg}} = 0.3$ and $\lambda_{\mathrm{adv}} = 0.7$, and the conditions with LPIPS or SAA add $\lambda_{\mathrm{lpips}} = 0.2$ or $\lambda_{\mathrm{SAA}} = 0.3$, respectively. Full training configuration and reproducibility details are in Appendix A; loss-formulation and weight-sweep analyses are in Appendix B.

Evaluation data and metrics. Evaluation uses a fixed set of $n = 1000$ images from SA-1B (Kirillov et al., 2023) (not used in training), with each image a 1280×1280 center crop. Image quality

Table 1: **Quality–robustness comparison at $\alpha = 0.2$.** SA-1B, $n = 1000$. NATIVE is Pixel Seal's existing fixed-JND attenuation, FREE is attenuation learned without LPIPS or SAA, SAA and LPIPS each apply only the corresponding regularizer, and LPIPS+SAA applies both. NATIVE is the released checkpoint's result; learned conditions are mean ± standard deviation over 3 seeds. All thirteen runs are evaluated together, so every condition uses the same image–message pairs; LPIPS uses the standard $[-1, 1]$ input (Appendix A).

Condition	PSNR↑	SSIM↑	LPIPS↓	Valuemetric(40)↑	Geometric(35)↑	Combined(3)↑
NATIVE	46.42	0.9972	0.0099	0.9839	0.9611	0.9405
FREE	41.05 ± 0.50	0.9957 ± 0.0003	0.0248 ± 0.0018	0.9942 ± 0.0001	0.9693 ± 0.0002	0.9786 ± 0.0004
SAA	42.02 ± 0.17	0.9950 ± 0.0001	0.0218 ± 0.0004	0.9941 ± 0.0001	0.9699 ± 0.0002	0.9796 ± 0.0008
LPIPS	42.09 ± 0.50	0.9965 ± 0.0005	0.0183 ± 0.0023	0.9932 ± 0.0005	0.9689 ± 0.0004	0.9756 ± 0.0018
LPIPS+SAA	43.10 ± 0.17	0.9960 ± 0.0001	0.0176 ± 0.0003	0.9928 ± 0.0002	0.9690 ± 0.0003	0.9755 ± 0.0005

is evaluated with PSNR, SSIM, and LPIPS (Zhang et al., 2018), and robustness is measured as the bit accuracy (the fraction of correctly recovered bits) of the message recovered after transformation. NATIVE is evaluated using the released checkpoint as-is, and the learned conditions report the mean and standard deviation over three distinct training seeds.

Attack configuration. Robustness evaluation is organized into three categories: Valuemetric, Geometric, and Combined. Valuemetric uses 40 attacks in total: 9 brightness, 9 contrast, 10 hue, 1 grayscale, 6 JPEG, and 5 Gaussian blur. Geometric uses 35 attacks in total: 1 identity, 1 horizontal flip, 5 rotation, 10 resize, 10 crop, and 8 perspective. Combined applies JPEG compression, center crop, and brightness adjustment in sequence, using three JPEG qualities (40, 60, 80); the center-crop ratio and brightness factor are fixed at 0.71 and 0.5, respectively. Each evaluation image is paired with one randomly sampled message, and the same image–message pairs are reused across the conditions being compared. Bit accuracy is averaged over the n=1000 images per attack, and each category score is the unweighted mean over its attacks.

4.2 QUALITY–ROBUSTNESS ANALYSIS

Comparison at a fixed global coefficient. In Table 1, replacing NATIVE (Fixed-JND) with the base FREE learned attenuation raises Combined accuracy from 94.05% to 97.86% (+3.81 pp), while PSNR drops from 46.42 to 41.05 dB (−5.37 dB). Both differences are significant under an image-level paired bootstrap (Appendix C). The other learned conditions likewise show higher bit accuracy and lower PSNR than NATIVE. The four learned conditions form a 2×2 ablation over the LPIPS loss and SAA: each regularizer raises PSNR relative to FREE by about 1 dB with only a small, sign-varying change in Combined accuracy, and combining them raises PSNR further (+2.05 dB over FREE) without a correspondingly large robustness cost (full per-condition deltas in Appendix B.1). The three quality metrics do not move together, however: adding SAA raises PSNR and lowers LPIPS but slightly *lowers* SSIM (exact values in Appendix B.1), so we describe its effect as improving PSNR and LPIPS specifically, not image quality in general. Part of the gap to NATIVE arises because α alone does not fix the residual's energy: the attenuation map's mean magnitude differs across conditions even at the same coefficient (0.036 for NATIVE, 0.065 ± 0.003 for FREE, 0.052 ± 0.001 for LPIPS+SAA, averaged over the same n=1000 images), so embedded distortion is not held constant. This motivates the matched-PSNR comparison below.

Comparison at matched PSNR. Because different conditions produce different actual distortion at a fixed α, we independently adjust each condition's α to match a target PSNR and re-compare robustness. For each condition and seed, we binary-search α until the evaluation set's mean PSNR is within 0.02 dB of the target, then evaluate robustness at that calibrated α on the same n=1000 images (targets 43–47.5 dB, from about the highest learned condition's PSNR at $\alpha = 0.2$ (43.10 dB) to above NATIVE's (46.42 dB), with 46.04 dB as an anchor near the latter). This comparison is therefore an oracle-normalized analysis of each method's quality–robustness curve, not a deployable operating point chosen without access to evaluation-time distortion. Matching *mean* PSNR fixes the geometric (not arithmetic) mean of per-image MSE and does not imply matched perceptual quality.

In Figure 2(a), over 43–46.04 dB, NATIVE shows 0.36–0.38 pp higher Combined accuracy than the best learned condition (and 0.9–2.2 pp higher than FREE and LPIPS), and at 47.5 dB it is within

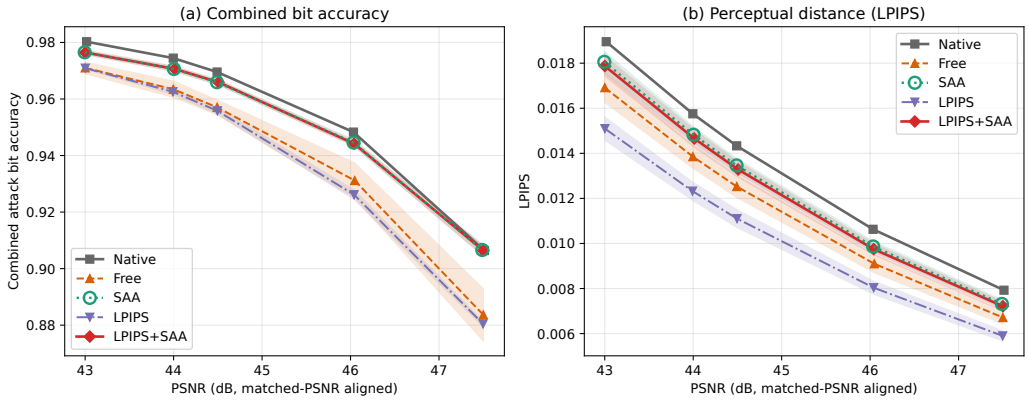


Figure 2: **Comparison at matched PSNR.** Each condition's α is independently calibrated to the target mean PSNR; all five conditions appear in both panels (SAA and LPIPS+SAA nearly coincide in (a)). (a) Combined-attack bit accuracy. (b) LPIPS (standard $[-1, 1]$ input). $n = 1000$ SA-1B images; shading is the sample standard deviation over seeds for learned conditions.

0.02 pp of the two SAA conditions; at 43–46.04 dB every seed-level paired interval against either SAA condition excludes zero (Appendix C). The robustness gain observed at fixed α therefore does not persist at matched PSNR; at all but the highest target it reverses.

At matched PSNR, the same 2×2 ablation separates cleanly. Adding SAA raises Combined accuracy by 0.55–2.29 pp without the LPIPS loss (SAA vs. FREE) and by 0.55–2.59 pp with it (LPIPS+SAA vs. LPIPS), the gain growing with the target PSNR and every seed-level interval excluding zero, so SAA and LPIPS+SAA are the two best learned conditions and lie within 0.04 pp of each other. Adding the LPIPS loss instead changes accuracy by at most 0.51 pp (almost always downward) but lowers LPIPS, and SAA's gain comes with 0.0013–0.0028 higher LPIPS (LPIPS+SAA vs. LPIPS).

In Figure 2(b), every learned condition shows lower LPIPS than NATIVE at matched PSNR, with LPIPS lowest (e.g., at 44 dB, 0.0123 vs. 0.0147 for LPIPS+SAA and 0.0158 for NATIVE), and higher SSIM at every target (e.g., 0.9967 for LPIPS+SAA vs. 0.9955 for NATIVE at 44 dB), a metric not used in training. This advantage is specific to matching PSNR, however.

Comparison at matched robustness. Reading the same curves at a common Combined accuracy of 95% (linear interpolation between the two nearest grid points), NATIVE reaches it at the highest PSNR (45.91 dB) and so with the least distortion, ahead of SAA and LPIPS+SAA (45.64 and 45.63 dB) and of FREE and LPIPS (44.92 and 44.80 dB), while LPIPS differs by at most 0.0011 across all five conditions (NATIVE 0.0109, LPIPS 0.0105). At equal robustness, then, learned attenuation offers no clear perceptual benefit for its extra training and $2.1\times$ embedding time; this is a single interpolated operating point, not a full equal-robustness curve.

Qualitative comparison of attenuation maps. Figure 3 shows that FREE, trained without explicit JND alignment, follows image structure as NATIVE's fixed JND map does but spreads its response more broadly into surrounding regions, and that SAA pulls that response back toward edges and textures. This is also reflected quantitatively: relative to the corresponding condition without it (FREE or LPIPS), adding SAA multiplies the attenuation map's high-frequency DCT magnitude fraction by 3.8–4.2$\times$ and its mean Sobel gradient magnitude by 2.6–2.7$\times$ – nearly matching NATIVE on the Sobel statistic and recovering about half of NATIVE's DCT magnitude fraction – while changing the map's mean magnitude by only about 10% in either case, consistent with the mean-normalized formulation of $\mathcal{L}_{\text{SAA}}$ (§3.2), which is invariant to a positive global rescaling of the learned map (full per-condition DCT/Sobel/magnitude values in Appendix A). These statistics summarize the attenuation map's own spatial structure, however, and do not establish that this structure directly causes the observed quality or robustness changes. Appendix G analyzes robustness under resolution reduction.



Figure 3: **Qualitative comparison of attenuation maps across scenes.** NATIVE shows a relatively localized response around edges and textures, whereas FREE shows a similar response spread more broadly into surrounding areas. Applying SAA reduces this spatial spread, and LPIPS+SAA likewise retains a similar localized structure.

4.3 EXPLORATORY ANALYSIS ON ADDITIONAL WATERMARKING BACKBONES

To examine whether these effects carry over to other watermarking backbones, we apply the same procedure to MaskWM (Hu et al., 2025) (both D_128 and ED_128 configurations), WAM (Sander et al., 2025), and VideoSeal (Fernandez et al., 2024), training only the attenuation module; NATIVE, FREE, and LPIPS+SAA are defined as in §3.2, with NATIVE denoting each backbone's default attenuation.

Cross-backbone analysis uses 36 attacks (29 Valuemetric, 7 Geometric) applicable to all backbones, with the same attack implementation (`videoseal.augmentation`, not a backbone's native noise layer); evaluation images and resolution differ by backbone (n=1000 each: the SA-1B set of §4.1 for Pixel Seal, COCO val2017 at 256×256 for MaskWM and WAM, SA-1B at 256×256 for VideoSeal), so conditions should be compared within a backbone. The additional backbones' learned conditions use one training seed each at a fixed global coefficient (each backbone's native operating point, hereafter *native-op*; Table 7), so robustness differences in that table cannot be separated from distortion level; matched-PSNR results are in Appendix E.1.

In Table 7 (Appendix E), the relationship among the three conditions differs by backbone: FREE shows the highest Common(36) but substantially lower PSNR for Pixel Seal, WAM, and VideoSeal, with LPIPS+SAA recovering part of the quality while retaining part of the gain; for MaskWM, LPIPS+SAA instead achieves higher PSNR and lower LPIPS than NATIVE at the cost of lower Common(36). This is exploratory evidence that learned attenuation shifts each backbone's operating point in a backbone-dependent direction, not evidence of a backbone-independent improvement or a ranking of methods, given the single seed and fixed coefficient per backbone; because evaluation data, resolution, base objective (e.g., MaskWM's FREE already includes an image-MSE term; only Pixel Seal and VideoSeal use $\mathcal{L}_{\text{adv}}$), and training budget also differ across backbones (Appendix A), we do not attribute the direction to the backbone architecture itself (each backbone's own noise layer: Appendix F). Part of these differences reflects unmatched distortion: matching PSNR (Appendix E.1) narrows the JPEG-only accuracy gap between NATIVE and the learned conditions from 2.8–10.0 pp at native-op to under 1.2 pp at every grid point tested for all three backbones, although on VideoSeal the learned conditions keep a small, significant lead over NATIVE, unlike on Pixel Seal. We use JPEG-only accuracy because the Combined attack is uninformative for MaskWM, and do not extend this result to Common(36).

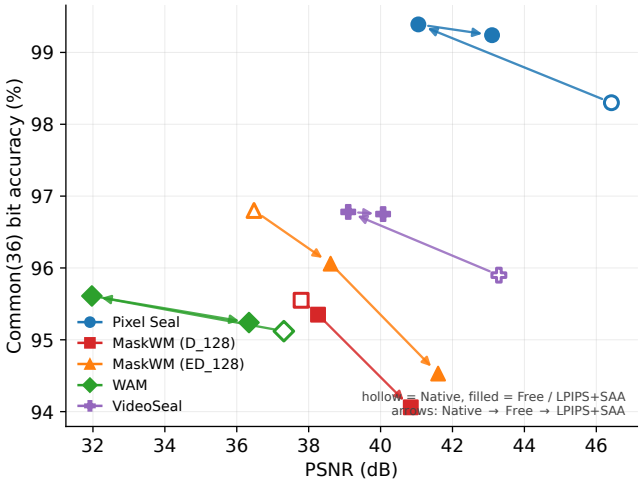


Figure 4: **Quality–robustness trajectory across watermarking backbones.** Each backbone's NATIVE, FREE, and LPIPS+SAA conditions (Table 7) are plotted as PSNR vs. Common(36), connected in that order. For Pixel Seal, WAM, and VideoSeal, FREE gains robustness at a large PSNR cost that LPIPS+SAA partially reverses; both MaskWM configurations instead move monotonically toward higher PSNR and lower Common(36).

Figure 4 shows these trajectories. As with Pixel Seal, LPIPS+SAA yields more localized boundary patterns than FREE on these backbones without becoming uniform (Figure 6 in Appendix D).

Limitations and future work. Our evaluation centers on COCO, SA-1B, and predefined image transformations; robustness to generative re-synthesis or video compression remains unverified. The learned attenuation module adds ∼116k parameters and more than doubles per-image embedding time relative to fixed JND (16.0 → 33.4 ms; Appendix A). The additional-backbone experiments use a single seed and add the same λ_{lpips} and λ_{SAA} as Pixel Seal to each backbone's own objective without per-backbone tuning, while the rest of the objective and the training schedule differ by backbone; their matched-PSNR results (Appendix E.1) cover JPEG attacks only. Perceptual quality is measured only with PSNR, SSIM, and LPIPS, without a human study. The four main-text hyperparameters ($\lambda_{\text{msg}}, \lambda_{\text{adv}}, \lambda_{\text{SAA}}, \lambda_{\text{lpips}}$) were each selected from a single-seed sweep (Appendix B), so the chosen values could reflect single-run noise; however, the sweeps used separate checkpoints evaluated on COCO, and re-evaluating all five headline conditions on a disjoint SA-1B shard ($n = 500$) reproduces Table 1's PSNR, SSIM, and robustness-over-NATIVE trends (Table 6).

5 CONCLUSION

Replacing Pixel Seal's fixed-JND attenuation with a learned module raised robustness at the cost of quality at a fixed coefficient; the LPIPS loss and SAA mitigated this drop in PSNR (+2.05 dB together) and LPIPS, though adding SAA slightly lowered SSIM. On Pixel Seal, once PSNR was matched, fixed JND took a small robustness lead at most targets; the learned conditions kept lower LPIPS and higher SSIM there, but at matched robustness fixed JND needed the least distortion and the LPIPS difference nearly vanished. Among the learned conditions, SAA accounted for the robustness differences at matched PSNR (+0.55–2.59 pp with or without the LPIPS loss), while the LPIPS loss mainly lowered LPIPS. On MaskWM, WAM, and VideoSeal, matching PSNR likewise narrowed the JPEG-attack gap to under 1.2 pp, although on VideoSeal the learned conditions kept a small lead. The main lesson is methodological: attenuation should be compared along PSNR-calibrated quality–robustness curves, since moving from a fixed coefficient to matched PSNR reversed the robustness ranking here.

AI USE STATEMENT

Generative AI tools (Claude, Anthropic; ChatGPT, OpenAI) were used in writing experiment code, verifying results, and drafting and proofreading the manuscript. All AI-proposed content was verified by the authors against actual execution results, original literature, and experimental data, and the authors are responsible for the final content and claims.

ETHICS STATEMENT

This work uses public image datasets and publicly released pretrained watermarking models. Invisible watermarking can be used for content-provenance tracking and copyright protection, but it can also be misused for covert tracking, or its detection results can be over-interpreted as conclusive evidence of provenance. Real-world deployment should therefore consider transparency, false-positive and false-negative rates, and the limits of attribution.

REPRODUCIBILITY STATEMENT

The experimental setup is described in §4.1, SAA is defined in §3, and Appendix A gives full checkpoint, training, attack-parameter, and metric-definition details for reproduction. Code reproducing the core method and main experiments is provided in anonymized form as supplementary material. As noted in §4.1, all learned conditions share the same batch size, schedule, and optimizer, and reported metrics use the final (10th) epoch checkpoint.

REFERENCES

Mahnoosh Bagheri, Majid Mohrekesh, Nader Karimi, and Shadrokh Samavi. Adaptive control of embedding strength in image watermarking using neural networks. *arXiv preprint arXiv:2001.03251*, 2020.

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

Runyi Hu, Jie Zhang, Shiqian Zhao, Nils Lukas, Jiwei Li, Qing Guo, Han Qiu, and Tianwei Zhang. Mask image watermarking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Jian Jin, Dong Yu, Weisi Lin, Lili Meng, Hao Wang, and Huaxiang Zhang. Full RGB just noticeable difference (JND) modelling. *arXiv preprint arXiv:2203.00629*, 2022.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. doi: 10.23915/distill.00003.

Jiayi Qin, Jingwei Li, and Chuan Wu. JND-guided neural watermarking with spatial transformer decoding for screen-capture robustness. *arXiv preprint arXiv:2603.26766*, 2026.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

Tom Sander, Pierre Fernandez, Alain Durmus, Teddy Furon, and Matthijs Douze. Watermark Anything with localized messages. In *International Conference on Learning Representations (ICLR)*, 2025.

Tomáš Souček, Pierre Fernandez, Hady Elsahar, Sylvestre-Alvise Rebuffi, Valeriu Lacatusu, Tuan Tran, Tom Sander, and Alexandre Mourachko. Pixel Seal: Adversarial-only training for invisible image and video watermarking. *arXiv preprint arXiv:2512.16874*, 2025.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations (ICLR) Workshop*, 2015.

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

Baowei Wang, Yufeng Wu, and Guiling Wang. Adaptor: Improving the robustness and imperceptibility of watermarking by the adaptive strength factor. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(11):6260–6272, 2023.

Miaohui Wang, Yukuan Zhu, Rong Zhang, and Wuyuan Xie. MetaJND: A meta-learning approach for just noticeable difference estimation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3151–3159, 2024.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations (ICLR)*, 2017.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Xiaohui Zhang, Weisi Lin, and Ping Xue. Just-noticeable difference estimation with pixels in images. *Journal of Visual Communication and Image Representation*, 19(1):30–41, 2008.

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

A Implementation and Reproducibility Details

Implementation details. The attenuation module is trained using distributed data-parallel training on two NVIDIA TITAN RTX (24GB) GPUs with a per-GPU batch size of 4 (total batch size 8). AdamW is used with its default hyperparameters (β_1=0.9, β_2=0.999, weight decay 0.01), except that the learning rate is linearly warmed up from 10^{-6} to 10^{-4} over the first 1,000 iterations and then decays to 10^{-6} following a cosine schedule; no gradient clipping or mixed-precision training is used. Each training image is first resized to a shorter side of 768 and center-cropped to 768×768; then, at each training iteration, the batch's height and width are independently drawn from $\{256, 288, \ldots, 768\}$ (step 32) and the batch is resized to that resolution via bilinear interpolation with antialiasing. Each of the three training seeds (§4.1) re-initializes the attenuation module's weights and reseeds the PyTorch/NumPy random state controlling data order and message sampling.

Checkpoints and frozen components. All Pixel Seal experiments start from the released Pixel Seal checkpoint (`https://dl.fbaipublicfiles.com/videoseal/pixelseal/checkpoint.pth`), whose model card specifies a 256-bit `unet_base_yuv_quant` embedder, a `convnext_tiny` extractor, an embedder/extractor processing resolution of 256, additive blending, a native global coefficient $\alpha = 0.2$, and `jnd_1_1` attenuation. The embedder and extractor are restored from that checkpoint and frozen. The patch discriminator D operates on a single (luma) input channel, is newly initialized rather than restored from the checkpoint, and is updated with the standard hinge loss alongside A_ϕ from the first iteration (AdamW, learning rate 10^{-4}, same schedule as A_ϕ); among the watermarking networks, only A_ϕ receives parameter updates. A_ϕ itself adds ∼116k parameters, with compute scaling linearly with pixel count (4.8 GFLOPs at 256×256, the smallest training resolution, to 119.3 GFLOPs at 1280×1280, the SA-1B evaluation resolution). Embedding times (§4.2, §4.3) are wall-clock times of the full embedding call (embedder, attenuation, and blending) for one 1280×1280 image (batch size 1) on an NVIDIA TITAN RTX, with CUDA synchronization before and after the call, averaged over the 1,000 evaluation images without a separate warm-up.

Additional backbones. MaskWM uses the official `D_128bits` and `ED_128bits` checkpoints at a pinned upstream commit, with message length 128 and that family's native JND factor (1.3 for D, 1.75 for ED). WAM uses the official released checkpoint at a pinned upstream commit; its message length (32) and native scalars (`scaling_w`= 2.0, `scaling_i`= 1.0, `jnd_1_3_blue` attenuation) are read from the checkpoint's own `params.json` rather than re-declared. VideoSeal uses this repository's `videoseal_1.0` baseline with message length 256 (native $\alpha = 0.2$, `jnd_1_1` attenuation); being Pixel Seal's predecessor in the same code base, it is trained with the Pixel Seal objective and weights ($\lambda_{\text{msg}} = 0.3$, $\lambda_{\text{adv}} = 0.7$, plus $\lambda_{\text{lpips}} = 0.2$, $\lambda_{\text{SAA}} = 0.3$ for LPIPS+SAA) but for 5 epochs of 1,000 iterations on COCO at a fixed 256×256, with a per-GPU batch of 32 on two GPUs, the same warmup-then-cosine schedule over 5 epochs, and the final (5th) epoch checkpoint. For MaskWM and WAM, the training objective is that backbone's own upstream loss plus $\lambda_{\text{lpips}} = 0.2$ and $\lambda_{\text{SAA}} = 0.3$ (LPIPS+SAA only), the same two weights used for Pixel Seal; FREE uses the upstream loss alone. For MaskWM this upstream loss already contains an image-reconstruction MSE term (encoder weight 1.0, alongside message weight 0.2 and mask weight 0.5, the converged values of its upstream schedule), so MaskWM's FREE condition is not free of a quality constraint, consistent with its higher PSNR than NATIVE in Table 7; WAM's upstream loss consists of detection and decoding terms only. Neither of these two backbones uses an adversarial discriminator term — MaskWM's objective has none, and the released WAM checkpoint's own `params.json` sets both its discriminator and perceptual weights to zero — so $\mathcal{L}_{\text{adv}}$ (Eq. 6) is specific to the Pixel Seal and VideoSeal settings. The MaskWM and WAM runs use 5,000 optimizer updates at an effective global batch of 32, AdamW (lr = 10^{-4}, weight decay 0.01, no schedule), at 256 × 256 on COCO 2017.

LPIPS configuration. $\mathcal{L}_{\text{lpips}}$ and the reported LPIPS both use the `lpips` package's VGG network. The training loss takes $[0, 1]$ inputs (the package default `normalize=False`), whereas all reported LPIPS values use the standard $[-1, 1]$ input at the full evaluation resolution.

Training-time attacks and non-differentiable transforms. Training augmentations come from `configs/all_augs.yaml`. At every iteration, two augmentations are drawn independently and applied in sequence; each of the 16 image-domain types carries selection weight 1, and the weights

are normalized, so the draw is uniform over the applicable types. If a drawn transform changes the spatial size, the result is bilinearly resized (with antialiasing) back to the input size before the next draw. The sampled parameter ranges are: resize and crop edge ratios in $[0.7, 1.5]$ and $[0.5, 1.0]$; rotation in $[-10°, 10°]$ plus $90°$ multiples; perspective distortion scale in $[0.1, 0.5]$; JPEG quality in $[40, 80]$; Gaussian blur kernel size in $[3, 17]$; median filter kernel size 3; brightness, contrast, and saturation factors in $[0.5, 2.0]$; and hue shift in $[-0.1, 0.1]$. The two non-differentiable transforms, JPEG and the median filter, are applied with a straight-through estimator: the forward value is the transformed image and the backward pass is the identity, implemented as $\mathcal{T}_{\mathrm{nd}}(x) \leftarrow \mathrm{sg}[\mathcal{T}_{\mathrm{nd}}(x) - x] + x$ with $\mathrm{sg}[\cdot]$ the stop-gradient operator. The JPEG transform itself clips to $[0, 1]$ and then encodes and decodes through a real 8-bit JPEG codec.

Evaluation attack parameters. The 78-attack suite of §4.1 uses the following parameter values. Valuemetric(40): brightness and contrast factors $\{0.1, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0\}$ (9 each); hue shifts $\{-0.4, -0.3, \ldots, 0.5\}$ (10); grayscale (1); JPEG qualities $\{40, 50, 60, 70, 80, 90\}$ (6); Gaussian blur kernel sizes $\{3, 5, 9, 13, 17\}$ (5). Geometric(35): identity (1); horizontal flip (1); rotations $\{5°, 10°, 30°, 45°, 90°\}$ (5); resize and crop ratios $\{0.32, 0.45, 0.55, 0.63, 0.71, 0.77, 0.84, 0.89, 0.95, 1.0\}$ (10 each); perspective distortion scales $\{0.1, 0.2, \ldots, 0.8\}$ (8). Combined(3): JPEG quality $\{40, 60, 80\}$ each composed with center crop 0.71 and brightness factor 0.5. The Common(36) subset used for cross-backbone comparison (Table 7) is the subset of these applicable to every backbone: 29 Valuemetric attacks (JPEG $\times 6$, Gaussian blur $\times 5$, brightness $\times 9$, contrast $\times 9$) and 7 Geometric ones (identity, horizontal flip, rotation $\times 5$).

Crop ratio convention. Crop and resize ratios are *edge-length* ratios, not area ratios: a crop at ratio r retains an $\lfloor rH \rfloor \times \lfloor rW \rfloor$ region. The Combined attack's crop at 0.71 therefore keeps about 50% of the image area.

Clipping and quantization. The watermarked image x_w is clipped to $[0, 1]$ before any attack is applied and before any quality metric is computed. No explicit 8-bit quantization step is applied to x_w outside the JPEG attacks, which pass through a real 8-bit codec; bit accuracy is therefore measured on floating-point images for the non-JPEG attacks.

Attenuation-map statistics. The DCT and Sobel statistics reported in §4.2 are computed on the single-channel attenuation map at the full evaluation resolution, on the map's own $[0, 1]$ scale, with no per-image normalization or rescaling. The high-frequency DCT magnitude fraction (a fraction of absolute coefficients, not of squared energy) applies a 2-D orthonormal DCT-II to the whole map and reports $\sum_{\mathcal{H}} |D_{u,v}| \,/\, \sum_{u,v} |D_{u,v}|$, where the high-frequency region is $\mathcal{H} = \{(u, v) : \sqrt{(u/H)^2 + (v/W)^2} \geq 0.5\}$, i.e. normalized radial frequency at or above half of the per-axis Nyquist frequency (the radius reaches about $\sqrt{2}$ at the highest-frequency corner). The Sobel statistic is the mean of $\sqrt{g_x^2 + g_y^2 + \varepsilon}$ over all pixels, with g_x, g_y the responses of the standard 3×3 Sobel kernels applied directly to the map. Per condition, averaged over the same $n{=}1000$ evaluation images: the DCT magnitude fraction is 0.537 for NATIVE, 0.069 ± 0.012 for FREE, 0.071 ± 0.010 for LPIPS, 0.293 ± 0.026 for SAA, and 0.271 ± 0.010 for LPIPS+SAA (sample standard deviation over seeds, as in Table 1); the mean Sobel gradient magnitude is 0.046, 0.021 ± 0.001, 0.018 ± 0.001, 0.056 ± 0.001, and 0.048 ± 0.001 in the same order (the last nearly matching NATIVE). The map's own mean magnitude, by contrast, changes little when SAA is added: $0.065 \rightarrow 0.059$ without LPIPS and $0.058 \rightarrow 0.052$ with it, versus 0.036 for NATIVE.

B LOSS-ABLATION AND HYPERPARAMETER-WEIGHT-SWEEP ANALYSIS

This appendix reports the ablation studies and hyperparameter weight sweeps performed during the design of our loss functions.

B.1 THE 2×2 COMBINATION OF $\mathcal{L}_{\mathrm{lpips}}$ AND $\mathcal{L}_{\mathrm{SAA}}$: DETAILED RESULTS

The independent contributions of $\mathcal{L}_{\mathrm{lpips}}$ and $\mathcal{L}_{\mathrm{SAA}}$ were first observed in an early exploratory experiment at $n = 200$ (COCO test2017, single seed) – the SAA condition ($\lambda_{\mathrm{SAA}} = 0.3$, $\lambda_{\mathrm{lpips}} = 0$)

jointly improved PSNR ($41.40 \rightarrow 41.96$ dB) and LPIPS ($0.0307 \rightarrow 0.0255$) over FREE, suggesting that $\mathcal{L}_{\mathrm{SAA}}$ contributes independently even without the image-quality loss. This preliminary result was subsequently reproduced in the same direction by the full 3-training-seed, SA-1B $n = 1000$ 2×2 factorial in Table 1 (§4.2), whose full per-condition deltas we report below.

LPIPS raises PSNR over FREE from 41.05 to 42.09 dB ($+1.03$ dB) and lowers LPIPS from 0.0248 to 0.0183, while Combined accuracy decreases from 97.86% to 97.56% (-0.30 pp). SAA raises PSNR from 41.05 to 42.02 dB ($+0.97$ dB) and lowers LPIPS from 0.0248 to 0.0218, while Combined accuracy increases from 97.86% to 97.96% ($+0.10$ pp). We interpret this accuracy change as an observation from a limited set of three seeds.

LPIPS+SAA raises PSNR over FREE by 2.05 dB and lowers LPIPS to 0.0176. Its Combined accuracy is 97.55%, 0.31 pp lower than FREE. Relative to LPIPS alone, adding SAA raises PSNR by 1.02 dB while lowering Combined accuracy by 0.01 pp. Comparing SAA's two PSNR deltas ($+0.97$ dB without LPIPS, $+1.02$ dB with LPIPS) shows a consistent main effect on PSNR regardless of LPIPS, whereas its Combined-accuracy delta changes sign ($+0.10$ vs. -0.01 pp), i.e., its small effect on robustness at a fixed coefficient has no consistent direction. In this setting, then, combining the two regularizers further improves the learned attenuation's PSNR and LPIPS while keeping the robustness change small. SSIM is the exception and moves in the opposite direction: adding SAA lowers it from 0.9957 to 0.9950 without LPIPS and from 0.9965 to 0.9960 with it (Table 1). We therefore state the improvement in terms of PSNR and LPIPS specifically rather than as a uniform gain in image quality.

The matched-PSNR counterpart of this ablation (§4.2) isolates SAA at a controlled distortion budget. Per target (43, 44, 44.5, 46.04, 47.5 dB), adding SAA raises Combined accuracy by 0.55, 0.72, 0.89, 1.34, and 2.29 pp without the LPIPS loss (SAA vs. FREE) and by 0.55, 0.81, 1.01, 1.81, and 2.59 pp with it (LPIPS+SAA vs. LPIPS); the latter raises LPIPS by 0.0028, 0.0023, 0.0022, 0.0017, and 0.0013 at the same targets.

Two design-choice controls for SAA. SAA makes two design choices: using a dense per-pixel attenuation map rather than a single scalar, and mean-normalizing both maps before aligning them (Eq. 8). We test each with a dedicated control under the same $n = 200$, single-seed, COCO protocol as above (conditions LPIPS+SAA and SAA re-evaluated together with the two controls for a shared reference point: PSNR 43.17/41.72 dB, Combined(J60) 95.93%/97.19%, DCT magnitude fraction 0.189/0.300, Sobel 0.045/0.067, LPIPS 0.0198/0.0268).

*Scalar-gating control (*SCALAR_SAA*).* We replace the U-Net attenuation module with $A(x) = s_\theta(x) \cdot \mathrm{JND}(x)$, where $s_\theta(x)$ is a single learned scalar per image applied to the fixed JND map, so the spatial shape is pinned to NATIVE's by construction. Its DCT magnitude fraction (0.551) matches NATIVE's on the same images (0.551) to three decimals, confirming this. Because $A(x)$ is spatially proportional to $\mathrm{JND}(x)$ for every $s_\theta(x) > 0$, the mean-normalized maps satisfy $\widetilde{A}(x) = \widetilde{J}(x)$ regardless of $s_\theta(x)$ when ϵ is ignored, so $\mathcal{L}_{\mathrm{SAA}}$ is zero up to ϵ for this parameterization and gives essentially no training signal; we train SCALAR_SAA with $\lambda_{\mathrm{SAA}} = 0$ (all other weights unchanged) and let the remaining losses select $s_\theta(x)$. SCALAR_SAA reaches Combined(J60) accuracy 96.49% against LPIPS+SAA's 95.93% ($+0.56$ pp; a single, non-bootstrapped $n = 200$ point), but with lower PSNR (42.21 vs. 43.17 dB) and worse LPIPS (0.0246 vs. 0.0198). At this single operating point (fixed coefficient, $n = 200$, one seed), the two therefore sit at different points of the quality–robustness trade-off: the per-image scalar is slightly more robust but embeds more distortion. Because quality–robustness curves can cross, this does not establish an advantage for either parameterization at matched PSNR, which we did not measure for this control.

*Unnormalized-alignment control (*SAA_UNNORM*).* Removing the mean normalization from Eq. 8 and applying $\mathrm{MSE}(A(x), \mathrm{JND}(x))$ directly was considered and set aside early in development, on the expectation that it would degenerate into a fixed copy of $\mathrm{JND}(x)$. The result here shows a different failure mode: rather than converging toward NATIVE's sharp structure, SAA_UNNORM produces the *smoothest* map of any condition tested (DCT magnitude fraction 0.072, Sobel 0.008, both below every other condition including NATIVE). Figure 5 shows this directly – its maps are visibly blurred relative to every other condition, not sharpened to match NATIVE. Its LPIPS (0.0289) is worse than LPIPS+SAA's (0.0198) despite a higher PSNR (45.03 dB), consistent with distortion spread more diffusely rather than concentrated where JND allows it. Its lower Combined(J60) accuracy (94.83%) is not by itself evidence of a worse design, since it also embeds less distortion at this coefficient. In

this tested setting, then, the unnormalized loss produced a smoother map and a less favorable result, rather than a copy of JND. Because the unnormalized and mean-normalized losses differ in scale and gradient magnitude, the same $\lambda_{\mathrm{SAA}} = 0.3$ does not impose an equivalent constraint on both (the same caveat as the MSE-versus-LPIPS comparison in Appendix B.2); we did not tune λ for this control, so this is not evidence that mean normalization is superior in general.

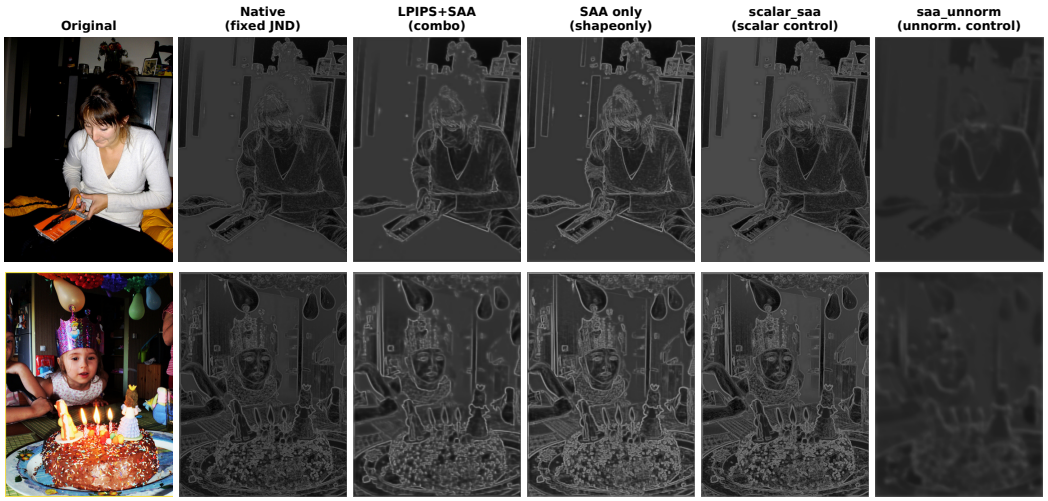


Figure 5: **Qualitative comparison of the two SAA controls.** Attenuation maps for two scenes, shown on a fixed raw scale. SCALAR_SAA reproduces NATIVE's spatial pattern exactly (by construction); SAA_UNNORM is visibly smoothed relative to every other condition, including NATIVE, rather than converging to it.

Quality at matched robustness. The matched-robustness comparison of §4.2 reads each condition's (PSNR, Combined accuracy, LPIPS) points off the matched-PSNR grid and interpolates linearly between the two grid points that bracket 95% Combined accuracy (44.5 and 46.04 dB for every condition), rather than running a new sweep. The resulting values are PSNR 45.91, 45.64, 45.63, 44.92, and 44.80 dB and LPIPS 0.0109, 0.0108, 0.0107, 0.0116, and 0.0105 for NATIVE, SAA, LPIPS+SAA, FREE, and LPIPS. They are interpolated, not independently measured at exactly 95%.

B.2 IMAGE-QUALITY LOSS FORMULATION ABLATION: LPIPS VS. MSE

To justify our choice of a deep perceptual loss ($\mathcal{L}_{\mathrm{lpips}}$) over simple pixel-wise mean squared error ($\mathcal{L}_{\mathrm{mse}} = \frac{1}{HW}\|x_w - x\|_2^2$) as the image-quality-preserving loss, we compared the two loss formulations at the same weight (0.2; $n = 200$, COCO test2017).

Table 2: **Comparison by image-quality loss formulation ($n = 200$, quality-loss weight 0.2).** Combined(J60) is bit accuracy under the JPEG60+Crop0.71+Brightness0.5 Combined attack; Mean A is the attenuation map's mean value. Bold: best among learned conditions.

Condition	PSNR (dB)↑	LPIPS↓	Mean A	Combined(J60)↑
NATIVE (Fixed-JND)	45.79	0.0127	0.0398	0.9015
FREE ($\lambda_{\mathrm{lpips}} = 0$)	41.40	0.0307	0.0652	0.9553
$\mathcal{L}_{\mathrm{mse}}$ (pixel MSE)	40.37	0.0353	0.0745	0.9658
LPIPS	**42.83**	**0.0189**	0.0559	0.9561

As shown in Table 2, $\mathcal{L}_{\mathrm{mse}}$ actually lowers PSNR relative to FREE, from 41.40 to 40.37 dB (-1.03 dB), and worsens LPIPS from 0.0307 to 0.0353, while inflating the mean attenuation-map value from 0.0652 to 0.0745. In contrast, $\mathcal{L}_{\mathrm{lpips}}$ lowers the mean attenuation-map value to 0.0559

while jointly improving PSNR and LPIPS. Based on this result, we use $\mathcal{L}_{\text{lpips}}$ as the perceptual-quality loss in subsequent experiments.

This comparison should be read narrowly. The two losses differ in numeric scale and in gradient magnitude, so the shared weight of 0.2 does not impose an equivalent effective constraint on both, and we did not sweep the weight separately for $\mathcal{L}_{\text{mse}}$. The result is therefore evidence that $\mathcal{L}_{\text{lpips}}$ was the more favorable of the two at the weight we tested — not evidence that a pixel-wise MSE quality loss is unsuitable in principle for this role.

B.3 Message-Recovery vs. Adversarial-Imperceptibility Weight ($\lambda_{\text{msg}}, \lambda_{\text{adv}}$) Sweep

We performed a 5-step sweep to determine the weight ratio between recovery and imperceptibility in the base training objective $\mathcal{L}_{\text{base}} = \lambda_{\text{msg}}\mathcal{L}_{\text{msg}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}$ ($n = 200$, COCO test2017, single training seed, Pixel Seal backbone). This sweep is a separate exploratory experiment from Table 2 and was run with an earlier version of the attenuation module (max-pooling downsampling and transposed-convolution upsampling, rather than the all-convolutional design of §3) and a different evaluation script. Absolute values are therefore not comparable across the two tables – e.g., `unet_D` reaches 39.41 dB, whereas FREE in Table 2, trained with the same loss weights, reaches 41.40 dB – and we use this sweep only for the relative trend across the $\lambda_{\text{msg}}/\lambda_{\text{adv}}$ ratio. Both this table and Table 2 report LPIPS with the standard $[-1, 1]$ input, but because the attenuation module and evaluation script differ, their absolute values are still not comparable.

Table 3: **Message-loss vs. adversarial-loss weight-sweep comparison ($n = 200$).** Combined is bit accuracy under the JPEG60+Crop0.71+Brightness0.5 attack; r_{JND} is the Pearson correlation with the Fixed-JND map.

Condition	λ_{msg}	λ_{adv}	PSNR↑	LPIPS↓	Combined↑	r_{JND}
NATIVE (Fixed-JND)	—	—	45.79	0.0127	89.8%	1.000
`unet_A`	1.0	0.1	38.64	0.0502	98.5%	0.615
`unet_B`	0.7	0.3	38.92	0.0482	98.5%	0.593
`unet_C`	0.5	0.5	39.15	0.0468	98.5%	0.561
`unet_D` (adopted)	0.3	0.7	39.41	0.0452	98.4%	0.539
`unet_E`	0.1	1.0	41.85	0.0471	96.8%	0.213

As λ_{adv} rises from 0.1 to 0.7, Combined-attack bit accuracy stays at a similar level of 98.4–98.5% while PSNR improves steadily from 38.64 to 39.41 dB. Moving further, to $\lambda_{\text{adv}} = 1.0$ ($\lambda_{\text{msg}} = 0.1$), Combined robustness drops to 96.8% (-1.6 pp), although PSNR rises to 41.85 dB; the map's correlation with JND also falls to 0.213. Considering the trade-off between robustness and image quality, we select $\lambda_{\text{msg}} = 0.3$, $\lambda_{\text{adv}} = 0.7$ as the standard baseline setting for all subsequent models.

B.4 Shape-Alignment (λ_{SAA}) and Perceptual (λ_{lpips}) Weight Sweeps

Shape-alignment weight (λ_{SAA}) sweep. With $\lambda_{\text{lpips}} = 0.2$ fixed, we sweep $\lambda_{\text{SAA}} \in \{0, 0.3, 0.5, 0.7\}$ ($n = 200$ COCO test2017 images, single training seed; the same evaluation batch as Table 2). Combined(J60) accuracy is 95.61% at 0, 96.22% at both 0.3 and 0.5, and 95.67% at 0.7, while PSNR stays within 42.83–43.43 dB. We adopt 0.3, the smallest weight reaching the highest Combined accuracy in this sweep.

Perceptual-quality weight (λ_{lpips}) sweep. A full sweep over $\lambda_{\text{lpips}} \in \{0.0, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7\}$ with $\lambda_{\text{SAA}} = 0$ (i.e., the LPIPS condition; $n = 1000$ COCO test2017 images, single training seed) uses training runs separate from, and earlier than, the headline checkpoints of Table 1. As λ_{lpips} increases from 0 to 0.7, PSNR rises from 41.45 to 44.52 dB and LPIPS falls from 0.0304 to 0.0117, while Combined(J60) accuracy declines: 96.03% at 0, 95.29% at 0.2, 94.24% at 0.3, 92.96% at 0.5, and 91.11% at 0.7. Because the decline steepens beyond 0.2 (1.05–1.85 pp per step versus under 0.4 pp per step below it), we select $\lambda_{\text{lpips}} = 0.2$ as the default, balancing quality and robustness; at that point the sweep gives PSNR 42.88 dB and LPIPS 0.0189.

C PAIRED BOOTSTRAP CONFIDENCE INTERVALS FOR THE FIXED-COEFFICIENT COMPARISON

The seed-averaged deltas of the four learned conditions against NATIVE in §4.2 (Table 1) are point estimates over $n = 1000$ SA-1B images. To quantify sampling uncertainty at the image level, we pair each image's per-seed value with NATIVE's value for that same image, average over the three seeds per image, and compute a nonparametric bootstrap over images. For 10,000 resamples (with replacement) of the 1,000 image indices, we recompute the mean paired difference and report the 2.5th/97.5th percentiles as a 95% confidence interval. This uncertainty is distinct from, and does not replace, the seed-level variability already reported as ± standard deviation in Table 1.

Table 4: **Image-level paired bootstrap 95% CIs against NATIVE ($n = 1000$ SA-1B images, 10,000 resamples).** Point estimates match the seed-averaged deltas reported in §4.2.

Condition	**ΔPSNR (dB)**, 95% CI	**ΔCombined (pp)**, 95% CI
FREE	-5.37, $[-5.44, -5.29]$	$+3.81$, $[+3.54, +4.08]$
SAA	-4.40, $[-4.42, -4.37]$	$+3.90$, $[+3.67, +4.15]$
LPIPS	-4.34, $[-4.41, -4.26]$	$+3.51$, $[+3.25, +3.77]$
LPIPS+SAA	-3.32, $[-3.34, -3.29]$	$+3.49$, $[+3.27, +3.72]$

All eight intervals in Table 4 exclude zero, so both the PSNR decrease and the Combined-accuracy increase at fixed $\alpha = 0.2$ are significant under this resampling procedure. This confirms that the fixed-α gap reported in §4.2 is not explained by image-sampling noise. It does not by itself address whether that gap persists under matched PSNR, which is the subject of the matched-PSNR comparison in §4.2 and Figure 2.

Paired bootstrap CIs at matched PSNR. We repeat the same procedure at the matched-PSNR grid for the comparisons underlying §4.2's claims (Table 5). Because NATIVE has no seed variants, we pair it against each learned seed separately; learned-versus-learned comparisons pair same-index seeds. The table gives, per target, the gap averaged over the three seed pairs and the lowest lower bound and highest upper bound among the three seed-level 95% intervals. At 43–46.04 dB every seed-level interval for NATIVE versus either SAA condition excludes zero; at 47.5 dB the mean gap is within 0.02 pp and, for each of the two comparisons, two of the three seed-level intervals include zero. All thirty intervals for SAA's own contribution, with or without the LPIPS loss, exclude zero, so this effect is small but consistent at matched PSNR, unlike its sign-varying effect at fixed α (Table 1).

Table 5: **Paired bootstrap at matched PSNR** (Combined accuracy, pp; $n = 1000$ SA-1B images, 10,000 resamples). Each cell: mean gap over the three seed pairs, then [lowest lower bound, highest upper bound] among the three seed-level 95% intervals.

Target	LPIPS+SAA − NATIVE	SAA − NATIVE	LPIPS+SAA − LPIPS	SAA − FREE
43 dB	-0.38 $[-0.61, -0.24]$	-0.37 $[-0.56, -0.24]$	$+0.55$ $[+0.34, +0.78]$	$+0.55$ $[+0.30, +0.90]$
44 dB	-0.38 $[-0.59, -0.21]$	-0.38 $[-0.58, -0.21]$	$+0.81$ $[+0.51, +1.12]$	$+0.72$ $[+0.37, +1.18]$
44.5 dB	-0.36 $[-0.57, -0.18]$	-0.36 $[-0.57, -0.18]$	$+1.01$ $[+0.61, +1.40]$	$+0.89$ $[+0.53, +1.35]$
46.04 dB	-0.41 $[-0.65, -0.19]$	-0.37 $[-0.64, -0.15]$	$+1.81$ $[+1.27, +2.37]$	$+1.34$ $[+0.57, +2.25]$
47.5 dB	$+0.00$ $[-0.42, +0.31]$	$+0.02$ $[-0.33, +0.38]$	$+2.59$ $[+1.87, +3.24]$	$+2.29$ $[+0.88, +3.45]$

Held-out re-evaluation. Table 6 re-evaluates the checkpoints of Table 1 on a disjoint SA-1B shard ($n = 500$, no overlap with the headline images), with all thirteen runs in one evaluation so that every condition uses the same image–message pairs. The main trends of Table 1 are reproduced: all learned conditions have higher robustness and lower PSNR than NATIVE; LPIPS and SAA each raise PSNR over FREE by about 1 dB and LPIPS+SAA by about 2 dB; adding SAA lowers SSIM; and Combined accuracy is highest for SAA and within 0.3 pp across the other learned conditions. The sign of SAA's small Combined-accuracy change in the presence of the LPIPS loss is not reproduced (-0.01 pp in Table 1, $+0.02$ pp here), consistent with its sign-varying effect at fixed α. LPIPS is omitted because this run used the non-standard input convention discussed in Appendix A.

Table 6: **Held-out re-evaluation** (disjoint SA-1B shard, $n = 500$, fixed $\alpha = 0.2$; learned conditions are mean $\pm$ standard deviation over 3 seeds).

Condition	PSNR↑	SSIM↑	Valuemetric(40)↑	Geometric(35)↑	Combined(3)↑
NATIVE	46.37	0.9973	0.9850	0.9607	0.9443
FREE	40.99 $\pm$ 0.51	0.9958 $\pm$ 0.0003	0.9951 $\pm$ 0.0001	0.9692 $\pm$ 0.0002	0.9800 $\pm$ 0.0005
SAA	41.96 $\pm$ 0.17	0.9951 $\pm$ 0.0001	0.9951 $\pm$ 0.0001	0.9704 $\pm$ 0.0002	0.9816 $\pm$ 0.0007
LPIPS	41.99 $\pm$ 0.50	0.9966 $\pm$ 0.0004	0.9942 $\pm$ 0.0004	0.9690 $\pm$ 0.0005	0.9773 $\pm$ 0.0019
LPIPS+SAA	43.04 $\pm$ 0.17	0.9961 $\pm$ 0.0001	0.9939 $\pm$ 0.0002	0.9691 $\pm$ 0.0001	0.9775 $\pm$ 0.0009

D QUALITATIVE COMPARISON OF ATTENUATION MAPS ACROSS OTHER BACKBONES (MASKWM, WAM, VIDEOSEAL)

This appendix provides a qualitative comparison of attenuation maps and watermark residuals for the four backbone configurations described in §4.3 (MaskWM D_128bits, MaskWM ED_128bits, WAM, VideoSeal (Fernandez et al., 2024)). Figure 6 compares, for the same representative scene (a motorcycle), the original image, each backbone's NATIVE fixed attenuation map, the FREE learned attenuation map, the LPIPS+SAA learned attenuation map, and the LPIPS+SAA watermark residual ($|x_w - x|$, $8\times$ magnified), side by side.

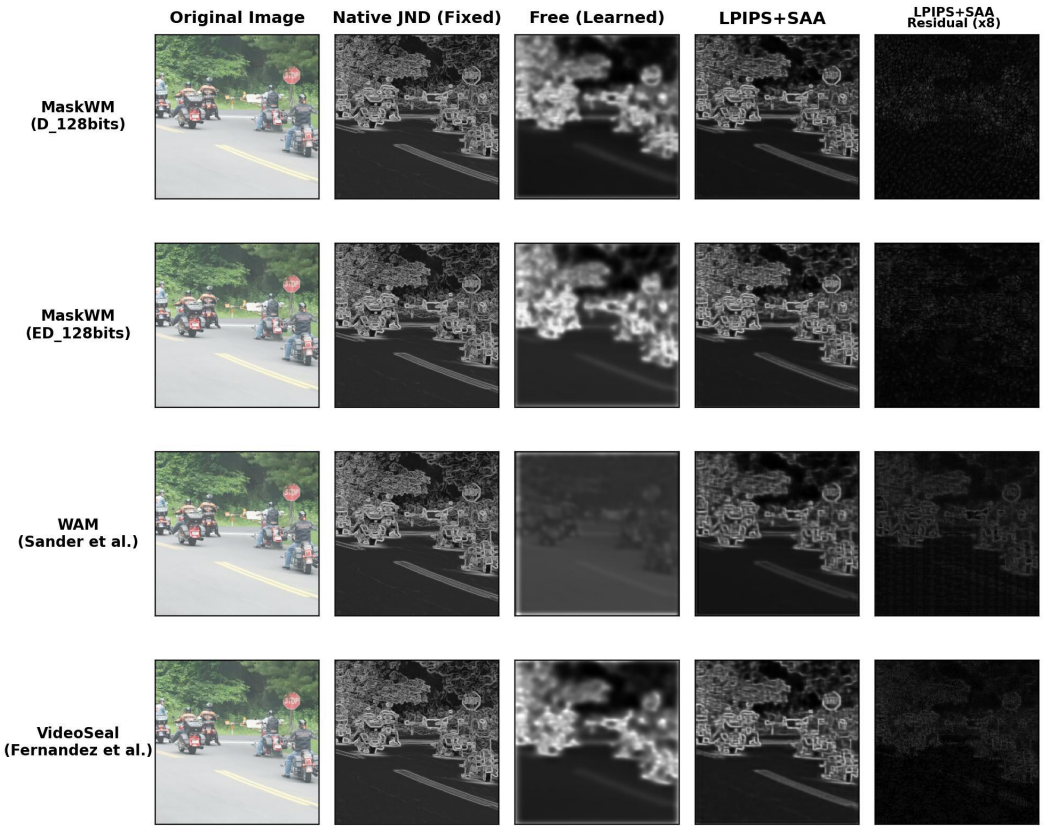


Figure 6: **Qualitative comparison of attenuation maps and watermark residuals across backbones.** Rows show, in order, MaskWM (D_128bits), MaskWM (ED_128bits), WAM (Sander et al., 2025), and VideoSeal (Fernandez et al., 2024). Columns show, in order, the original image, the NATIVE fixed attenuation map, the FREE learned attenuation map, the LPIPS+SAA learned attenuation map, and the LPIPS+SAA watermark residual ($|x_w - x|$, $8\times$ magnified). FREE and LPIPS+SAA show different spatial patterns across backbones, and LPIPS+SAA more clearly shows a localized boundary pattern similar to the native JND. LPIPS+SAA does not collapse into a uniform map on any backbone, whereas WAM's FREE map is nearly flat.

In Figure 6, NATIVE, FREE, and LPIPS+SAA show different attenuation patterns. FREE's maps smooth out fine boundaries and textures, whereas LPIPS+SAA shows a more localized boundary pattern across several backbones. This visualization is a qualitative result showing per-condition spatial differences; it does not imply that any particular pattern is the direct cause of robustness or perceptual-quality changes.

For the same Pixel Seal scene as Figure 3 in §4.2, a close-up comparison of the watermark residual itself ($|x_w - x|$) more directly shows how the difference in attenuation maps is reflected in the actual embedding pattern.

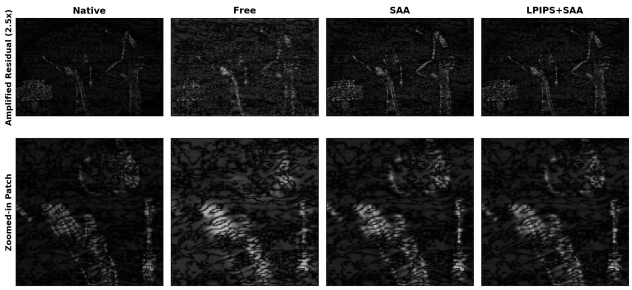


Figure 7: **Watermark residual comparison.** Residual ($|x_w - x|$, $2.5\times$ magnified) for the smooth-background scene of Figure 3; columns: NATIVE, FREE, SAA, LPIPS+SAA. Top: full residual. Bottom: enlarged patches. LPIPS+SAA and SAA show smaller residual magnitude than FREE in smooth background regions.

E CROSS-BACKBONE COMPARISON

This appendix gives the full Common(36) results summarized in §4.3 (Table 7) and their matched-PSNR counterpart for MaskWM, WAM, and VideoSeal (§E.1).

Across all four backbones, attenuation is injected with the same functional form, $x_w = x + \alpha \cdot A_\phi(x) \odot (\hat{x}_w - x)$, where $\hat{x}_w$ is that backbone's own unscaled pre-attenuation output (for Pixel Seal and VideoSeal, $\hat{x}_w = x + w$ with w as in Eq. 1, so this reduces to Eq. 2; for MaskWM and WAM, produced by a bridge that intercepts each backbone's built-in JND/attenuation call site and substitutes $A_\phi(x)$ for it), α is the same global coefficient introduced in §3, and A_ϕ is the identical U-Net attenuation architecture from §3, retrained from scratch per backbone. For MaskWM and WAM, the bridge additionally keeps each backbone's fixed per-channel weights on the residual ($[0.75, 0.5, 1.0]$ and $[0.5, 0.5, 1.0]$ in RGB). Each backbone's own message length, native input resolution, and pre/post-processing are otherwise unchanged from its official released checkpoint; only the attenuation call site is replaced.

E.1 MATCHED-PSNR COMPARISON FOR ADDITIONAL BACKBONES

The Common(36) comparison in Table 7 uses a single fixed global coefficient per backbone, so, as in the main text's Pixel Seal analysis (§4.2), part of the difference between NATIVE, FREE, and LPIPS+SAA may reflect unmatched embedding strength rather than the attenuation method itself. We repeat the matched-PSNR procedure from §4.2 for MaskWM (D_128bits, ED_128bits) and WAM (the same n=1000 COCO val2017 images at 256×256 and single training seed as Table 7), and for VideoSeal (n=1000 SA-1B images at 256×256, the same single training seed). We report JPEG-only bit accuracy (mean over JPEG qualities 40, 60, and 80, without crop or brightness) rather than the Combined attack used in Table 1: under this evaluation setup, MaskWM's Crop(0.71) distortion collapses decoding to near-chance accuracy for every condition, including NATIVE (Table 9, where we also note that this may reflect our preprocessing and crop implementation rather than the backbone itself), so the compound Combined attack is uninformative for this backbone specifically; using JPEG-only instead keeps the metric meaningful and comparable across all backbones in this appendix. Consequently, the narrowing reported here is a statement about JPEG robustness only, and we do not extend it to Common(36) or to the Combined attack.

Table 7: **Comparison on Pixel Seal and additional watermarking backbones.** Conditions are as in Table 1; for the additional backbones, NATIVE is that backbone's default attenuation. $n = 1000$ per backbone: SA-1B at 1280×1280 (Pixel Seal), COCO val2017 at 256×256 (MaskWM, WAM), and SA-1B at 256×256 (VideoSeal); compare conditions within a backbone. Common(36) is the mean over all 36 attacks (not the mean of the two category scores); additional backbones report an exploratory result from 1 training seed at a fixed global coefficient. All backbones share the attack implementation and report LPIPS with the standard $[-1, 1]$ input; VideoSeal's rows come from a separate evaluation script, on the same images as Table 11. Because quality and robustness are not compared at matched PSNR here, per-column values are not bolded.

Backbone	Condition	PSNR↑	LPIPS↓	Valuemetric(29)↑	Geometric(7)↑	Common(36)↑
Pixel Seal	NATIVE	46.42	0.0099	0.9805	0.9935	0.9830
	FREE	41.05	0.0248	0.9930	0.9974	0.9939
	LPIPS+SAA	43.10	0.0176	0.9912	0.9976	0.9924
MaskWM (D_128)	NATIVE	37.79	0.0581	0.9731	0.8827	0.9555
	FREE	38.26	0.0592	0.9706	0.8826	0.9535
	LPIPS+SAA	40.84	0.0323	0.9565	0.8745	0.9406
MaskWM (ED_128)	NATIVE	36.48	0.0577	0.9873	0.8879	0.9680
	FREE	38.61	0.0458	0.9785	0.8864	0.9606
	LPIPS+SAA	41.60	0.0206	0.9621	0.8757	0.9453
WAM	NATIVE	37.31	0.0399	0.9944	0.7722	0.9512
	FREE	31.97	0.1282	0.9993	0.7775	0.9561
	LPIPS+SAA	36.34	0.0453	0.9955	0.7736	0.9524
VideoSeal	NATIVE	43.29	0.0171	0.9830	0.8595	0.9590
	FREE	39.10	0.0550	0.9937	0.8602	0.9678
	LPIPS+SAA	40.07	0.0354	0.9932	0.8611	0.9675

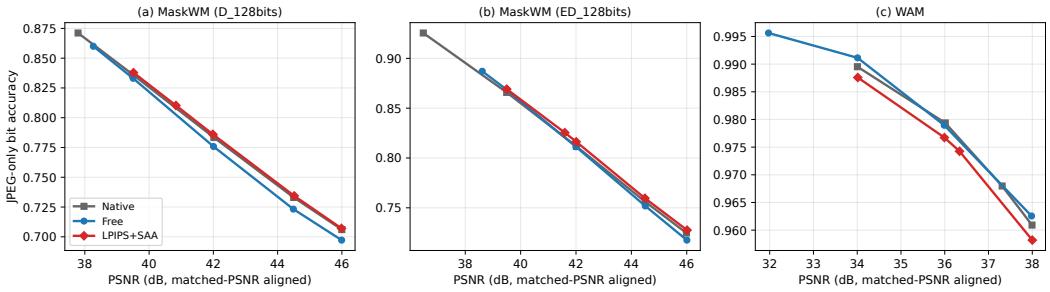


Figure 8: **Matched-PSNR comparison for MaskWM and WAM.** JPEG-only bit accuracy (mean over JPEG qualities 40, 60, 80) vs. PSNR, following the same calibration procedure as Figure 2. Each line contains that condition's native-op result (Table 7) together with its matched-PSNR grid targets; the native-op point is the leftmost one only when that condition's native-op PSNR lies below the lowest grid target (39.5 dB for MaskWM), which is not the case for MaskWM's LPIPS+SAA.

Figure 8 shows the result. As with Pixel Seal (§4.2), the native-op spread between conditions narrows substantially once PSNR is matched. For MaskWM D_128bits, native-op JPEG-only accuracy spans 6.09 pp across the three conditions (0.8103–0.8711), but at every one of the four matched-PSNR grid points (39.5, 42.0, 44.5, 46.0 dB) the spread is 0.47–1.11 pp. MaskWM ED_128bits shows the same pattern at a larger native-op scale (10.00 pp narrowing to 0.33–1.00 pp at matched PSNR). WAM's native-op spread was already comparatively small in bit-accuracy terms (2.76 pp) despite a large PSNR spread (31.97–37.31 dB, with FREE's LPIPS about three times higher: 0.128 vs. 0.040 and 0.045); at matched PSNR this narrows further to 0.26–0.44 pp, consistent with WAM's quality–robustness difference across conditions being concentrated in PSNR/LPIPS rather than in robustness itself. All results are from a single training seed per backbone, the same caveat as Table 7 (§4.3).

VideoSeal is reported separately in Table 8, using the same calibration procedure and metric. Its native-op JPEG-only gap to NATIVE (+2.91 pp for FREE, +2.50 pp for LPIPS+SAA) narrows to 0.23–0.99 pp and 0.07–0.53 pp across the four matched-PSNR targets (38.5–43.5 dB). Unlike on Pixel Seal (§4.2), the learned conditions keep a small lead at every target, and the paired bootstrap

95% CIs (over images) exclude zero at each point; the lead grows toward higher PSNR, where JPEG accuracy is further from saturation. As with the other backbones, this is a single training seed, JPEG attacks only, and without an LPIPS measurement at matched PSNR.

Table 8: **Matched-PSNR comparison for VideoSeal** (n=1000 SA-1B images at 256×256, single training seed). JPEG-only bit accuracy (%, mean over JPEG qualities 40, 60, 80) at each condition's calibrated α; achieved PSNR is within 0.02 dB of each target. Δ is the paired difference from NATIVE in pp, with a 95% paired bootstrap CI over images (10,000 resamples). The native-op row uses each condition's default $\alpha = 0.2$ (achieved PSNR 43.29 / 39.10 / 40.07 dB).

Target PSNR	NATIVE	FREE	Δ_{FREE}	LPIPS+SAA	$\Delta_{\text{LPIPS+SAA}}$
native-op	96.50	99.41	+2.91 [2.77, 3.04]	99.00	+2.50 [2.40, 2.61]
38.5 dB	99.26	99.48	+0.23 [0.19, 0.26]	99.32	+0.07 [0.04, 0.09]
40.0 dB	98.91	99.23	+0.32 [0.27, 0.37]	99.02	+0.11 [0.09, 0.14]
41.5 dB	98.19	98.67	+0.48 [0.41, 0.55]	98.44	+0.25 [0.21, 0.28]
43.5 dB	96.21	97.20	+0.99 [0.89, 1.09]	96.74	+0.53 [0.48, 0.58]

F EVALUATION UNDER EACH BACKBONE'S OWN OFFICIAL NOISE LAYER

As described in §4.3, the Common(36) evaluation in Table 7 uses a common attack implementation across backbones, but it does not cover each backbone's own specialized training-time noise (e.g., MaskWM's Gaussian noise (GN), salt-and-pepper noise (SP), and CropResize). Tables 9, 10, and 11 therefore measure NATIVE, FREE, and LPIPS+SAA under each backbone's own noise layer, using the distortion types in MaskWM's (Hu et al., 2025), WAM's (Sander et al., 2025), and VideoSeal's (Fernandez et al., 2024) official noise layers that correspond to attack types in our 78-attack suite (§4.1), plus median filtering and saturation for VideoSeal, which the suite lacks; all are evaluated at $n = 1000$ on SA-1B. This is a separate evaluation from Table 7 for MaskWM and WAM: the tables here use the SA-1B images of §4.1, whereas Table 7 uses COCO val2017 images at 256×256 (a seed-21 sample after reserving 256 images for hyperparameter calibration), which is why, e.g., MaskWM D_128bits' NATIVE PSNR differs between the two (37.79 dB in Table 7 vs. 37.45 dB here).

Table 9: **Detailed evaluation under MaskWM's own official noise layer ($n = 1000$, SA-1B).** Bit accuracy and PSNR comparison on distortions overlapping with Table 1 (compression (JPEG), geometric distortion (Rotate, CropResize, Perspective, HFlip)).

	MaskWM D_128bits			**MaskWM ED_128bits**		
Distortion (official noise_layers)	**NATIVE**	**FREE**	**LPIPS+SAA**	**NATIVE**	**FREE**	**LPIPS+SAA**
PSNR (dB)↑	37.45	37.96	40.49	36.13	38.25	41.23
Clean (no distortion)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
JPEG	0.8367	0.8182	0.7648	0.9145	0.8586	0.7840
Rotate	0.8742	0.8737	0.8639	0.8918	0.8906	0.8761
CropResize	0.5158	0.5135	0.5052	0.5207	0.5182	0.5132
Perspective	0.9587	0.9585	0.9373	0.9837	0.9816	0.9594
HorizontalFlip	1.0000	1.0000	1.0000	1.0000	1.0000	0.9997

As shown in Table 9, MaskWM maintains 100% bit accuracy on Clean, and near-100% on HorizontalFlip, even under its own official noise layer, but JPEG shows a substantial spread (76.5–91.5% across conditions) rather than saturating near 100%. LPIPS+SAA is consistently below NATIVE on JPEG, by the largest margin of any distortion in this table (−7.2 pp on D_128bits, −13.0 pp on ED_128bits). It is also below NATIVE on Perspective by a smaller but non-trivial margin (−2.1 pp on D_128bits, −2.4 pp on ED_128bits), and by a smaller margin still on Rotate (−1.0/−1.6 pp) – an imperceptibility–robustness trade-off observed across the non-saturated geometric and compression distortions in this table, most pronounced for JPEG and, to a lesser extent, Perspective. CropResize (an official distortion defined by the backbone's own noise layer) shows accuracy near 0.50 — chance — across all conditions including NATIVE. We report this as a crop vulnerability observed for MaskWM *under this evaluation setup*, not as an intrinsic property of the backbone:

our pipeline applies the distortion through its own preprocessing and crop implementation rather than MaskWM's official end-to-end inference procedure, and a difference in either could produce the same symptom. This caveat also applies to the near-chance crop result reported again in Appendix E.1. LPIPS+SAA improves PSNR by $+3.04$ dB on D_128bits and $+5.10$ dB on ED_128bits.

Table 10: **Detailed evaluation under WAM's own official noise layer ($n = 1000$, SA-1B).** Bit accuracy and PSNR comparison on distortions overlapping with Table 1 (compression (JPEG), filtering (GF), color transforms (Brightness, Contrast), geometric distortion (Rotate, Crop, Resize, Perspective, HFlip), and compound distortion (CropResizePad)).

Distortion (official noise_layers)	**NATIVE**	**FREE**	**LPIPS+SAA**
PSNR (dB)↑	37.01	32.14	36.05
Clean (no distortion)	1.0000	1.0000	0.9999
JPEG ($Q = 70$)	0.9931	0.9992	0.9946
JPEG ($Q = 50$)	0.9673	0.9957	0.9749
GF (Gaussian Blur 5×5)	0.9999	1.0000	0.9999
Brightness ($\times 1.5$)	0.9994	0.9997	0.9995
Contrast ($\times 1.5$)	0.9998	0.9999	0.9998
Rotate (10°)	0.9981	0.9992	0.9985
Crop ($r = 0.7$)	0.9934	0.9941	0.9937
Resize ($r = 0.7$)	0.9999	0.9999	0.9999
Perspective ($s = 0.2$)	0.9973	0.9985	0.9978
HorizontalFlip	0.9999	1.0000	0.9999
CropResizePad	0.8396	0.8426	0.8422

Table 10 shows results under WAM's official augmentation pipeline (`watermark_anything.augmentation`). The FREE condition, lacking an imperceptibility loss, learns an excessive embedding strength, causing PSNR to drop from 37.01 to 32.14 dB, whereas LPIPS+SAA restores PSNR to 36.05 dB, recovering most of the gap to NATIVE's PSNR. At the same time, LPIPS+SAA matches or exceeds NATIVE bit accuracy on every listed distortion except Clean and GF, including JPEG ($0.9931 \rightarrow 0.9946$ at $Q = 70$, $0.9673 \rightarrow 0.9749$ at $Q = 50$), Crop ($0.9934 \rightarrow 0.9937$), Perspective ($0.9973 \rightarrow 0.9978$), and the compound geometric distortion CropResizePad ($0.8396 \rightarrow 0.8422$). The two exceptions are lower by under 0.0001 (Clean $1.0000 \rightarrow 0.9999$; GF $0.99991 \rightarrow 0.99988$ unrounded), a negligible difference. Overall, LPIPS+SAA recovers most of the quality lost by FREE while matching or slightly exceeding NATIVE's bit accuracy under WAM's own noise layer. These are fixed-coefficient results at a 0.96 dB lower PSNR than NATIVE, however, so – as with Pixel Seal (§4.2) and WAM's own matched-PSNR comparison (Appendix E.1) – the small accuracy margins should not be read as a robustness advantage at matched PSNR.

Table 11 presents results (SA-1B, $n = 1000$) under VideoSeal's official augmentation pipeline (`videoseal.augmentation`). As with the other backbones, the FREE condition's excessive embedding strength causes PSNR to drop sharply by 4.19 dB, from 43.29 to 39.10 dB, whereas LPIPS+SAA partially recovers, gaining $+0.97$ dB to reach 40.07 dB. In terms of robustness, LPIPS+SAA consistently exceeds NATIVE across every distortion category, including the official compound distortion Compound ($0.8110 \rightarrow 0.9088$, $+9.78$ pp), JPEG ($0.9881 \rightarrow 0.9974$ at $Q = 70$, $0.9559 \rightarrow 0.9887$ at $Q = 50$), Median Filter ($0.9622 \rightarrow 0.9810$), and Crop ($0.9687 \rightarrow 0.9763$). It further exceeds even FREE (which embeds at a lower PSNR) on Crop, Rotate, Brightness, Contrast, and MF. These are fixed-coefficient results in which LPIPS+SAA embeds at a 3.22 dB lower PSNR than NATIVE, and under matched PSNR, VideoSeal's JPEG-only gap between LPIPS+SAA and NATIVE shrinks from 2.50 pp to 0.07–0.53 pp (Table 8); we therefore do not read these margins as a robustness advantage of this size at matched PSNR. We report them as per-distortion empirical results and do not, from this evaluation alone, attribute the cause to any particular spatial structure or frequency characteristic.

G DETAILED ROBUSTNESS ANALYSIS AT EXTREMELY LOW RESOLUTION

In deployment settings such as social-media uploads, messenger transmission, and web thumbnail generation, watermarked high-resolution images can be downscaled substantially, to 512×512,

Table 11: **Detailed evaluation under VideoSeal's own official noise layer ($n = 1000$, SA-1B).** Bit accuracy and PSNR comparison for Clean, compression (JPEG), filtering (GF, MF), color transforms (Brightness, Contrast, Saturation), geometric distortion (Rotate, Crop, Resize, Perspective, HFlip), and compound distortion (Compound: JPEG60+Crop0.71+Brightness0.5).

Distortion (official noise_layers)	NATIVE	FREE	LPIPS+SAA
PSNR (dB)↑	43.29	39.10	40.07
Clean (no distortion)	0.9997	0.9998	0.9999
JPEG ($Q = 70$)	0.9881	0.9984	0.9974
JPEG ($Q = 50$)	0.9559	0.9939	0.9887
GF (Gaussian Blur 5×5)	0.9995	0.9998	0.9999
MF (Median Filter 5×5)	0.9622	0.9797	0.9810
Brightness ($\times 1.5$)	0.9779	0.9837	0.9853
Contrast ($\times 1.5$)	0.9917	0.9951	0.9958
Saturation ($\times 1.5$)	0.9993	0.9997	0.9998
Rotate (10°)	0.9875	0.9957	0.9966
Crop ($r = 0.7$)	0.9687	0.9708	0.9763
Resize ($r = 0.7$)	0.9990	0.9997	0.9998
Perspective ($s = 0.2$)	0.9993	0.9997	0.9998
HorizontalFlip	0.9997	0.9998	0.9999
Compound (J60+Crop+B0.5)	0.8110	0.9245	0.9088

256×256, or lower. This appendix analyzes how watermark decoding accuracy changes under such resolution reduction.

G.1 COMBINED-ATTACK ROBUSTNESS TREND UNDER RESOLUTION REDUCTION

We measured bit-accuracy trends on the SA-1B dataset ($n = 1000$, mean over 3 training seeds) while progressively downscaling image resolution across six steps from 1024 to 32 ($1024, 512, 256, 128, 64, 32$).

Table 12: **Bit-accuracy trend under resolution reduction ($n = 1000$, SA-1B, fixed $\alpha = 0.2$; LPIPS+SAA is the mean over 3 seeds).** Valuemetric(40), Geometric(35), and Combined(3) are the mean bit accuracy over 40 valuemetric, 35 geometric, and 3 combined attacks, respectively. Values are not bolded because the two conditions embed at different distortion levels (see §G.2).

Resolution	NATIVE			LPIPS+SAA		
(px)	Valuemetric(40)	Geometric(35)	Combined(3)	Valuemetric(40)	Geometric(35)	Combined(3)
1024×1024	0.9832	0.9632	0.9332	0.9926	0.9705	0.9727
512×512	0.9781	0.9627	0.8357	0.9904	0.9720	0.9363
256×256	0.9502	0.9386	0.5754	0.9698	0.9564	0.6778
128×128	0.7544	0.7340	0.4998	0.8122	0.7804	0.5001
64×64	0.5915	0.5644	0.5008	0.6078	0.5753	0.5006
32×32	0.5079	0.5061	0.5007	0.5113	0.5080	0.5001

Table 12 shows that, at fixed α, NATIVE degrades faster than LPIPS+SAA as resolution decreases. In particular, at a practically relevant intermediate resolution of 256×256, NATIVE's Combined-attack accuracy decreases to 0.5754, whereas LPIPS+SAA records 0.6778, a $+10.24$ pp difference. At 512×512 as well, LPIPS+SAA maintains a Combined-attack accuracy of 0.9363, $+10.06$ pp over NATIVE's 0.8357.

However, this gap is not unique to LPIPS+SAA; it is a phenomenon observed across learned attenuation in general. At the same two resolutions, FREE's Combined-attack accuracy is 0.7264 at 256×256 and 0.9499 at 512×512 – 1.4–4.9 pp higher than LPIPS+SAA's (0.6778/0.9363), with LPIPS in between (0.6977/0.9372). That is, all three learned conditions lead NATIVE by a wide margin at this fixed coefficient. The ordering among them, FREE $>$ LPIPS $\geq$ LPIPS+SAA, is consistent with the main-text observation (§4.2) that at fixed α the image-quality loss trades a small amount of robustness for lower distortion. We therefore interpret the relative low-resolution robustness as a property of the learned attenuation family in general rather than an effect specific to

LPIPS+SAA. The comparison above, however, fixes $\alpha = 0.2$ for every condition, so it does not separate the attenuation map's spatial design from the larger residual the learned conditions embed at that coefficient; §G.2 addresses this only for a resize-only attack at 64–128 px.

G.2 Matched-PSNR Comparison at Extremely Low Resolution

As in the main text (§4.2), the conditions in Table 12 embed at different distortion levels at a fixed α (46.42 dB for NATIVE versus 41.36–43.30 dB for the single-seed learned runs used below), so we repeat the matched-PSNR procedure of §4.2 along the resolution axis. To isolate resolution reduction itself, we use a single resize attack rather than the attack suite of Table 12: the watermarked 1280×1280 image is downscaled to $R \times R$ and the message is decoded directly from the downscaled image. Calibration and evaluation share the same n=1000 SA-1B images and messages, following §4.2. Unlike Table 12, this experiment uses one training seed per learned condition rather than a three-seed average.

We run the comparison in both directions. *Forward* lowers each learned condition's α until its mean PSNR matches NATIVE's native-op 46.42 dB; *reverse* raises NATIVE's α until its mean PSNR matches each learned condition's native-op value. Every calibrated point lands within 0.016 dB of its target. Only 64 and 128 px are informative: at 256 px and above every condition and setting falls between 0.9974 and 0.9999, and at 32 px every condition falls between 0.512 and 0.528, at or near the chance level of 0.5. We therefore restrict the analysis to 64 and 128 px.

Table 13: **Matched-PSNR comparison under resolution reduction ($n = 1000$, SA-1B, one training seed per learned condition).** Bit accuracy when the watermarked 1280×1280 image is downscaled to 64×64 or 128×128 and decoded directly. In the forward block, Δ is the row's accuracy minus that of NATIVE at native α (first row), so a positive Δ favors the learned condition; in the reverse block, Δ is the matched NATIVE's accuracy minus that of the corresponding learned condition at native α, so a positive Δ favors NATIVE. Resolutions outside 64–128 px are omitted: 256 px and above are saturated (0.9974–0.9999) and 32 px is near chance (0.512–0.528) for every condition.

Condition	Setting	α	PSNR (dB)	LPIPS	64 px	Δ_{64}	128 px	Δ_{128}
Forward: each learned condition's α lowered to NATIVE*'s PSNR*								
NATIVE	native α	0.2000	46.42	0.0099	0.6605	–	0.9034	–
FREE	native α	0.2000	41.36	0.0234	0.6898	+0.0293	0.9681	+0.0647
FREE	matched	0.1115	46.42	0.0085	0.6465	−0.0141	0.8886	−0.0148
LPIPS	native α	0.2000	42.42	0.0165	0.6787	+0.0182	0.9583	+0.0549
LPIPS	matched	0.1259	46.43	0.0071	0.6410	−0.0195	0.8821	−0.0214
LPIPS+SAA	native α	0.2000	43.30	0.0174	0.6822	+0.0217	0.9577	+0.0543
LPIPS+SAA	matched	0.1395	46.42	0.0093	0.6534	−0.0071	0.9013	−0.0021
Reverse: NATIVE*'s α raised to each learned condition's PSNR*								
NATIVE	matched to FREE	0.3598	41.34	0.0260	0.7039	+0.0141	0.9761	+0.0080
NATIVE	matched to LPIPS	0.3179	42.41	0.0213	0.6969	+0.0182	0.9685	+0.0102
NATIVE	matched to LPIPS+SAA	0.2873	43.29	0.0181	0.6903	+0.0081	0.9607	+0.0030

Table 13 shows that the low-resolution advantage of learned attenuation does not survive PSNR matching. At 128 px and native α, the learned conditions lead NATIVE by +0.0647 (FREE), +0.0549 (LPIPS), and +0.0543 (LPIPS+SAA); once their α is lowered to NATIVE's PSNR, those margins become −0.0148, −0.0214, and −0.0021, that is, reversed for FREE and LPIPS and essentially eliminated for LPIPS+SAA. The 64 px column behaves the same way: +0.0293, +0.0182, and +0.0217 at native α become −0.0141, −0.0195, and −0.0071. The reverse direction agrees. Raising NATIVE's α to each learned condition's PSNR at native α makes NATIVE slightly outperform that condition at 128 px (+0.0080, +0.0102, +0.0030) and at 64 px (+0.0141, +0.0182, +0.0081); NATIVE does not merely catch up but moves marginally ahead. Both directions agree that, for the resize-only attack at 64–128 px, the robustness advantage does not survive matching mean PSNR, in the same direction as the matched-PSNR result for the Combined attack in §4.2. We did not repeat this for the Combined attack at 256–512 px, where Table 12 shows the largest fixed-α gaps, so this result does not by itself explain those gaps. Among the learned conditions, LPIPS+SAA is the least affected: it has both the smallest matched deficit in the forward direction (−0.0021 at 128 px) and the smallest NATIVE lead in the reverse direction (+0.0030). It nonetheless retains no advantage in either direction, so this ordering does not change the conclusion.

As in §4.2, matching PSNR does not match perceptual distortion. At the forward matched points, all three learned conditions show lower LPIPS than NATIVE at native α (0.0085, 0.0071, and 0.0093 versus 0.0099), and, in the reverse direction, NATIVE's LPIPS at the matched point (0.0260, 0.0213, 0.0181) exceeds the corresponding learned condition's LPIPS at native α (0.0234, 0.0165, 0.0174). Again, what the calibration equalizes is the evaluation set's mean PSNR, which corresponds to the geometric rather than the arithmetic mean of the per-image MSEs, and which constrains neither the spatial nor the perceptual distribution of the residual. A difference of this kind is therefore expected; it indicates that the two families distribute a similar average distortion differently, and it does not restore a bit-accuracy advantage at either resolution.