Supplementary Material for Gaussian Image Steganography via Parameter-Domain Keyed Embeddings

Tong Wu[*]
East China University of Science and Technology
Shanghai, China
24012920@mail.ecust.edu.cn

Runze Cheng[†]
University of Cambridge
Cambridge, United Kingdom
rc2062@cam.ac.uk

Xiaoyue Fan
University College London
London, United Kingdom
merryxyfan@gmail.com

Kaan Akşit
University College London
London, United Kingdom
kaanaksit@kaanaksit.com

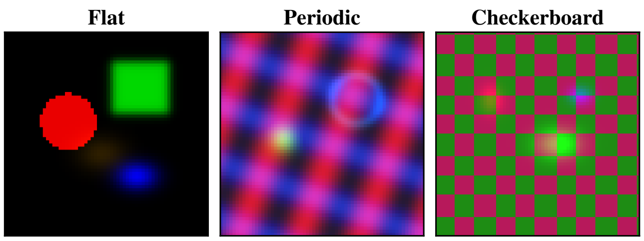

Figure S1: The three synthetic target images used in our evaluation: a flat gradient, a periodic stripe pattern, and a checkerboard pattern.

S1 Gaussian Parameter Distributions

Supplementary Figure S2 shows the fitted parameter distributions that guide carrier-channel selection.

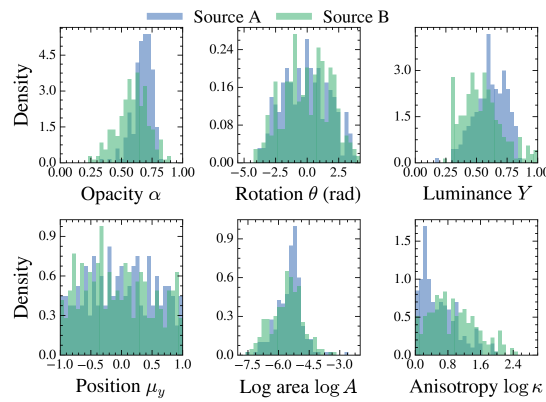

Figure S2: Gaussian parameter distributions. Qualitative distributions for the two source images in Figure 2 of the main paper. Opacity, color luminance, and log-anisotropy are used for the payload; rotation, position, and log-area are diagnostic only.

S2 Implementation Details

Embedding checks. A pilot check measures correct-key recovery, wrong-key behavior, and Peak Signal-to-Noise Ratio (PSNR) drop. If it fails, we reduce the edit budget and remove more low-reliability assignments. For periodic images, optimization starts from the initial parameter update when it satisfies the region limits. We retain the best checkpoint that passes these checks.

Extraction rule. For each four-bit block, the assignment map stores the candidate actions, parity matrix, and whitening bits. An action $a = (i, \ell, d, \gamma)$ specifies the Gaussian index i, channel ℓ, direction $d \in \{-1, 1\}$, and strength multiplier γ. Define $q_i^{\kappa} = \log \kappa_i$, $q_i^{\alpha} = \alpha_i$, and $q_i^{Y} = 0.299R_i + 0.587G_i + 0.114B_i$. With channel step δ_ℓ, extraction computes

$$s_a = \frac{d[q_i^{\ell}(\widetilde{P}) - q_i^{\ell}(\widehat{P})]}{\delta_\ell \max(\gamma, 10^{-3})}, \qquad \widehat{m}_a = \mathbf{1}[s_a > 0.5].$$

The score compares the observed change with the prescribed magnitude in direction d. The decision $\widehat{m}_a$ is one when more than half of that change is observed. Stacking these decisions gives $\widehat{\mathbf{m}}$. The parity mapping converts them into four message bits:

$$\hat{\mathbf{b}}_{\text{block}} = (H\widehat{\mathbf{m}} \bmod 2) \oplus \mathbf{w}.$$

Here H has one row per output bit and one column per candidate action. Each row selects decisions to sum modulo two. The stored whitening bits $\mathbf{w}$ are applied by bitwise XOR ($\oplus$): one flips the corresponding output bit and zero leaves it unchanged. They are all zero when whitening is disabled. Block outputs are concatenated and truncated to the payload length.

Periodicity score. The three factors are repetition intensity r_i, direction alignment a_i, and normalized edge support e_i, as in the main text. Let $\mathbf{f} = (f_x, f_y)$ be the dominant frequency from the fast Fourier transform (FFT) of the luminance image, in cycles per pixel, and let $\theta_f = \operatorname{atan2}(f_y, f_x)$. The angle $\theta_{\text{major},i}$ is the Gaussian major-axis angle defined in the main text. The implementation uses normalized Gaussian centers μ_i directly in the following calculation:

$$r_i = |\cos(2\pi\mu_i^T \mathbf{f})|, \qquad a_i = \frac{1 + \cos 2(\theta_{\text{major},i} - \theta_f)}{2},$$

$$e_i = \frac{E_i}{\max(\max_j E_j, 10^{-6})}, \qquad \phi_i = r_i a_i e_i.$$

Here E_i is the finite-difference gradient magnitude of the luminance image, sampled at the nearest pixel to the Gaussian center. Thus a_i measures major-axis alignment with $\mathbf{f}$, and e_i measures local edge strength relative to the largest sampled value. The main-text percentile rules use ϕ_i to assign regions. The transition test additionally admits non-core Gaussians with $e_i \geq 0.65$ and $r_i \geq 0.45$.

[*]Corresponding author.
[†]Also with Computational Light Laboratory, University College London.

Table S1: Correct-key BER and rendered-image distortion after adding noise to payload-carrying parameters. Each row averages 27 evaluations.

	Uniform		Gaussian	
Scale	BER_{ck}	ΔPSNR	BER_{ck}	ΔPSNR
0.005	0.000	0.568	0.060	0.561
0.010	0.042	0.589	0.130	0.589
0.020	0.069	0.674	0.144	0.721
0.050	0.194	1.136	0.273	1.419

S3 Optimization Objective

We use the five loss groups in Eq. (4) of the main paper. The fidelity term is image MSE. The message term combines correct-key binary cross-entropy with confidence margins for action decisions and decoded bits. The imperceptibility term combines parameter-distribution matching with image-detail and stripe-structure penalties. The key term promotes correct-key confidence and wrong-key outputs near 0.5. The auxiliary term limits changes in sensitive parameters and tests decoding after parameter noise, quantization, and 10% importance pruning. The robustness term $\mathcal{L}_{\text{hard}}$ applies a penalty to decoded values below 0.55 after parameter noise. For non-periodic images, three settings apply increasingly conservative edits; one is selected using the checks in Section S2. Periodic images use a single setting.

S4 Parameter-Noise Scale Sweep

We test four scales of uniform and Gaussian noise on the selected carrier parameters. For uniform noise, the scale is the half-width s of $U(-s, s)$; for Gaussian noise, it is the standard deviation s of $\mathcal{N}(0, s^2)$. For each scale, we evaluate nine fitted Gaussian sets (three targets × three fitting seeds) with three independent noise samples, giving 27 evaluations. The main paper reports the $s = 0.01$ uniform and $s = 0.005$ Gaussian settings.

S5 Comparison with Other Methods

We evaluate official pretrained RoSteALS (Bui et al., 2023) without retraining on the same 112 clean-fit renders. Its fixed, error-corrected 100-bit payload differs from our 8 bits; we compare decoded first bytes, and this payload difference limits the fidelity comparison. First-byte recovery is 109/112, versus 106/112 for all 100 bits. PSNR and SSIM use the same definitions and covers, measuring embedding distortion, not fit quality.

3D-GSW (Jang et al., 2025) extracts from 3D Gaussian renderings; GaussianMarker (Huang et al., 2024) extracts from both renderings and Gaussian parameters. In our method, decoding uses the embedded parameters, the corresponding clean reference parameters, and an assignment map produced by the encoder under key k. Payloads and distortion references differ, so their reported recovery and PSNR values are not directly comparable.

S6 Payload Capacity

Figure S3 reports the existing nine-fit development results for 4-, 8-, and 12-bit requests. All runs are included, including failed recovery or image-quality checks. All nine 16-bit requests were shortened to 8 bits, so none demonstrates 16-bit recovery; these runs are omitted from the curve and retained in the accompanying data.

Table S2: Byte recovery and cover fidelity on the same 112 covers. Recovery requires all eight bits to be correct.

Method	Recovered ↑	BER ↓	PSNR (dB) ↑	SSIM ↑
RoSteALS	109/112	0.003	$26.77_{\pm 2.75}$	$0.863_{\pm 0.051}$
Ours	**112/112**	**0.000**	$\mathbf{57.24}_{\pm 2.52}$	$\mathbf{0.999}_{\pm 0.001}$

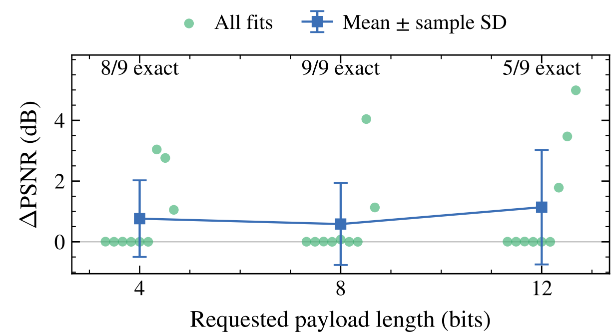

Figure S3: Capacity results on three synthetic targets with three fitting seeds each. Points show all nine runs per length; the line and bars show the mean and sample standard deviation. Labels give exact recovery counts. The vertical axis is the nonnegative PSNR drop relative to the clean fit.

S7 Cost-Weight Sensitivity

Nine newly generated fits cover three synthetic targets (seeds 0–2; 64 × 64; 256 Gaussians). We embed `01101111` for 200 iterations at learning rate 0.01, with hardening noise ratios 0.30/0.50. Starting from $(w_s, w_d, w_v, w_t) = (0.4, 0.3, 0.2, 0.1)$, we scale one weight by 0.75 or 1.25 and renormalize. We rerun all nine conditions on each clean fit (81 runs), holding its baseline-selected embedding settings fixed; periodic targets use the Section S3 setting.

All conditions recover 9/9 payloads with zero correct-key BER; selected checkpoints pass the recovery, wrong-key, and PSNR checks in Section S2. No run has payload fallback or extraction failure. These results concern local weight changes on synthetic fits.

Table S3: Nonnegative target-referenced PSNR drop from the clean fit (dB): mean ± sample standard deviation over nine fits. RGB is clamped to $[0, 1]$; weights are renormalized.

Weight varied	ΔPSNR (dB)	
	×0.75	×1.25
Baseline	$0.398_{\pm 0.791}$	
w_s	$0.338_{\pm 0.606}$	$0.596_{\pm 1.051}$
w_d	$0.446_{\pm 0.745}$	$0.498_{\pm 0.777}$
w_v	$0.483_{\pm 0.806}$	$0.169_{\pm 0.254}$
w_t	$0.412_{\pm 0.743}$	$0.316_{\pm 0.720}$