Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Diffusion-enhanced DCT steganography: reward-guided cover generation for increased embedding capacity with exact lossless-channel recovery

MD Ezazul Haque Mahi
BRAC University
Imraj Rabbani
BRAC University
Mustavi Hassan
BRAC University
Samia Tabassum Chowdhury
BRAC University
Md. Fahmid-Ul-Alam Juboraj
BRAC University
Muhammad Iqbal Hossain
iqbal.hossain@bracu.ac.bd

BRAC University

Article

Keywords: image steganography, discrete cosine transform, diffusion models, low-rank adaptation, cover image generation, steganalysis

Posted Date: November 16th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11590881/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Diffusion-enhanced DCT steganography: reward-guided cover generation for increased embedding capacity with exact lossless-channel recovery

MD Ezazul Haque Mahi[1], Imraj Rabbani[1], Mustavi Hassan[1], Samia Tabassum Chowdhury[1], Md. Fahmid-Ul-Alam Juboraj[1] and Muhammad Iqbal Hossain[1,*]

[1]Department of Computer Science and Engineering, BRAC University, Kha 224 Pragati Sarani, Merul Badda, Dhaka 1212, Bangladesh

[*]Corresponding author: iqbal.hossain@bracu.ac.bd

Abstract

Discrete cosine transform (DCT)-domain steganography embeds data into quantised DCT coefficients, but its capacity depends on how many cover coefficients are large enough to carry a bit. We propose Diffusion-Enhanced DCT Steganography (DEDS), which fine-tunes a generator to produce covers with more embeddable coefficients. Stable Diffusion v1.5 is fine-tuned with Low-Rank Adaptation, a differentiable DCT reward and an anchor loss. The payload is written by a margin-projected parity embedder whose expected distortion, $(\frac{1}{2}+\mu)^3Q^2/3$, and guaranteed coefficient-domain tolerance, μQ, we derive. On 1,000 prompt- and seed-matched 256×256 cover pairs, fine-tuning raised the mean number of eligible coefficients per 8×8 block from 1.57 to 16.64 and raw slot capacity from 0.056 to 0.418 bits per pixel (bpp) on the 566 pairs with cover-derived budgets. At 0.30 to 0.60 bpp every message was recovered exactly, with a luminance peak signal-to-noise ratio (PSNR) of 39.9 dB on fine-tuned covers, 6.8 dB above bin-centre embedding and above baseline covers in every payload bin. An optional redundancy mode survived mild JPEG compression and noise at up to 0.02 bpp. Rich-model steganalysis caught the four content-adaptive schemes (area under the curve, AUC, 0.85 to 0.99) and DEDS (AUC 0.997), so the method's value lies in capacity and fidelity, not security.

Keywords: image steganography; discrete cosine transform; diffusion models; low-rank adaptation; cover image generation; steganalysis

Introduction

Steganography hides data in ordinary media and is used in multimedia security, covert communication and digital forensics [1]. Images are the usual carrier, since they are everywhere on communication networks and can hold a lot of data [2]. Cryptography scrambles a message so that it cannot be read without a key; steganography instead hides the fact that a message exists at all [1]. An ideal steganography system is imperceptible, secure, robust and high in capacity. These properties work against each other, and no existing system has all four.

Methods work either in the spatial domain or in a transform domain [2]. Least significant bit (LSB) substitution, the classic spatial method, holds a lot of data but breaks under JPEG compression or resizing [2]; newer variants change fewer pixels while raising capacity and quality [3]. Transform domain methods modify coefficients of the discrete cosine transform (DCT) or the discrete wavelet transform instead [4]. The DCT, introduced by Ahmed, Natarajan and Rao in 1974 [5], is the transform behind JPEG and the one used here. Content-adaptive schemes such as HUGO [6], WOW [7] and UNIWARD [8] put their changes in textured regions. They are judged against

classifiers trained to tell cover from stego images, from feature-based detectors [9–11] to deep residual networks [12], so a modern scheme has to get past a detector built to find it as well as past the human eye.

Generative models offer another route. Studies have explored diffusion-based methods such as CRoSS [13] and GSD (StegoDiffusion) [14], which embed secrets in the diffusion process itself. Methods built on Stable Diffusion map the message into the latent noise and aim at lossy channels, including real social networks [15, 16], and PSyDUCK hides data in the denoising trajectory of a latent diffusion model without retraining it [17]. Other work embeds in existing images: GAN-based methods improve security in the JPEG domain [18] or refine DCT-embedded images with an encoder and decoder [19], and BAFLE-DCT picks DCT coefficients in regions of low saliency [20]. A third line of work changes the cover instead of the embedder, by selecting covers that suit the embedding [21].

When the cover is an existing image, or one picked from a collection, nothing shapes its frequency content for the payload it will carry. For quantised-DCT embedding this matters, because capacity depends on how many coefficients are large enough to carry a bit. That dependence is well known, and we do not claim it as a finding. To our knowledge, no study has tested whether a generator can be trained to supply such coefficients, or what that does to capacity, distortion, cover statistics and detectability. Diffusion models can already be fine-tuned toward differentiable or black-box rewards such as aesthetic or text-alignment scores [22, 23]. We apply the same idea with a reward defined on the block DCT of the generated image.

Diffusion-Enhanced DCT Steganography (DEDS) generates covers with a fine-tuned Stable Diffusion model [24] and hides data in their quantised DCT coefficients. The diffusion model only makes the cover. A separate embedder writes the payload afterwards, which sets DEDS apart from methods that hide data inside the diffusion process. The work makes three contributions.

First, a reward-guided objective for cover generation. A LoRA [25] adapter is trained with a differentiable reward, computed on a one-step estimate of the clean image, that favours DCT coefficients meeting the embedder's eligibility rule, while an anchor term keeps the model close to the pre-trained one. It raised the mean eligible count from 1.6 to 16.6 per block.

Second, a margin-projected parity embedder with an analytical distortion model. Instead of moving each quantised DCT coefficient to the centre of its bin, the embedder projects it onto the part of the target bin that lies at least μQ from either edge, so a coefficient that already has the right parity moves only if it sits near an edge. We derive the expected squared change, $(\frac{1}{2} + \mu)^3 Q^2/3$, and show that every written bit survives any later change smaller than μQ, so a single parameter trades distortion against tolerance. The rule is related to distortion-compensated quantisation index modulation [26], and every prediction is checked numerically (Supplementary Note 3). It gave 6.8 dB higher luminance PSNR than bin-centre writing.

Third, a controlled evaluation on 1,000 prompt- and seed-matched cover pairs, covering capacity, distortion, cover spectra, detectability with a detector trained for each embedder, and recovery after lossy processing.

The goal is more capacity and better image quality, with exact extraction when the stego image is stored without loss. We measure security against steganalysis and robustness to image processing but do not claim either. In the main mode the receiver needs only the stego image, with no key, model or side file.

Results

Evaluation corpus

Two cover-generation configurations are compared throughout: an unmodified Stable Diffusion v1.5 baseline, and a fine-tuned configuration using the LoRA-adapted U-Net trained with the combined diffusion, DCT reward, and anchor objective (Fig. 1a). Both are sampled with identical prompts and seeds, differing only in the trained adapter weights. The corpus contains 1,000 pairs of baseline and fine-tuned covers at 256×256 pixels, one pair for each prompt in a fixed bank of 1,000 prompts built from 100 texture subjects $\times$ 10 photographic modifiers (Methods). Every cover passes through the identical downstream pipeline (Fig. 1b).

Fine-tuning increases the number of embeddable DCT coefficients

A coefficient is eligible if its position belongs to the fixed index set $\mathcal{M}$ of 39 positions per block and its quantised magnitude is at least 2 (Methods). The per-block eligible count gives the cover-level embeddability statistic $\bar{e}$, an image average measured before embedding. Fine-tuning raised $\bar{e}$ from 1.57 ± 1.40 to 16.64 ± 2.60 (mean $\pm$ s.d., $n = 1{,}000$; Table 1), an increase observed in all 1,000 pairs (paired Wilcoxon signed-rank test, $p = 3.3 \times 10^{-165}$). The coefficient of variation of $\bar{e}$ fell from 0.89 to 0.16, so the fine-tuned generator delivers embeddable structure more consistently across prompts. The fraction of block-DCT energy at positions with $3 \leq u + v \leq 10$ rose from 0.051 to 0.354.

The set $\mathcal{M}$ is defined by the quantisation table (positions with step $Q \geq 8$) and covers the upper-frequency half of the block, including the highest-frequency position $(7, 7)$; it is not a mid-frequency band in the usual sense. Fig. 2a,b shows where the gain occurs: in baseline covers no position in $\mathcal{M}$ is eligible in more than 17% of blocks, whereas in fine-tuned covers the positions nearest the anti-diagonal are eligible in 62 to 82% of blocks and the highest-frequency positions in 8 to 14%.

Table 1. Cover statistics before embedding ($n = 1{,}000$ baseline/fine-tuned pairs, 256×256). Values are mean $\pm$ s.d. unless stated. $\bar{e}$, mean number of eligible coefficients per 8×8 block (of 39 positions in $\mathcal{M}$). K, adaptive position parameter (Eq. 10); when $\lfloor \bar{e} \rfloor = 0$ the embedder substitutes a fixed fallback $K = 24$ that is not derived from the cover. Raw slot capacity is $1022\,(K+1)/65536$ bits per pixel (Eq. 11). Eligible-slot capacity counts only the slots at the cover's own K (fallback included) that hold an eligible coefficient.

Statistic	Baseline	Fine-tuned
Block-DCT energy ratio, $3 \leq u + v \leq 10$	0.051 ± 0.030	0.354 ± 0.066
Eligible coefficients per block, $\bar{e}$	1.57 ± 1.40	16.64 ± 2.60
median (range)	1.17 (0.00 to 10.75)	16.68 (6.74 to 24.81)
Covers with $\lfloor \bar{e} \rfloor = 0$ (fallback K applied)	434 of 1,000	0 of 1,000
K, all covers (fallback included)	11.88 ± 10.74	23.92 ± 3.94
K, cover-derived only	2.59 ± 2.14 ($n = 566$)	23.92 ± 3.94 ($n = 1{,}000$)
Raw slot capacity (bpp), all covers	0.201 ± 0.167	0.389 ± 0.061
Raw slot capacity (bpp), cover-derived only	0.056 ± 0.033 ($n = 566$)	0.389 ± 0.061 ($n = 1{,}000$)
range, cover-derived only	0.031 to 0.250	0.156 to 0.577
Raw slot capacity (bpp), the same 566 pairs	0.056 ± 0.033	0.418 ± 0.050
Covers with cover-derived capacity ≥ 0.30 bpp	0 of 566	924 of 1,000
Eligible-slot capacity (bpp), all covers	0.011 ± 0.010	0.216 ± 0.046
Eligible-slot capacity (bpp), the same 566 pairs	0.013 ± 0.013	0.238 ± 0.039

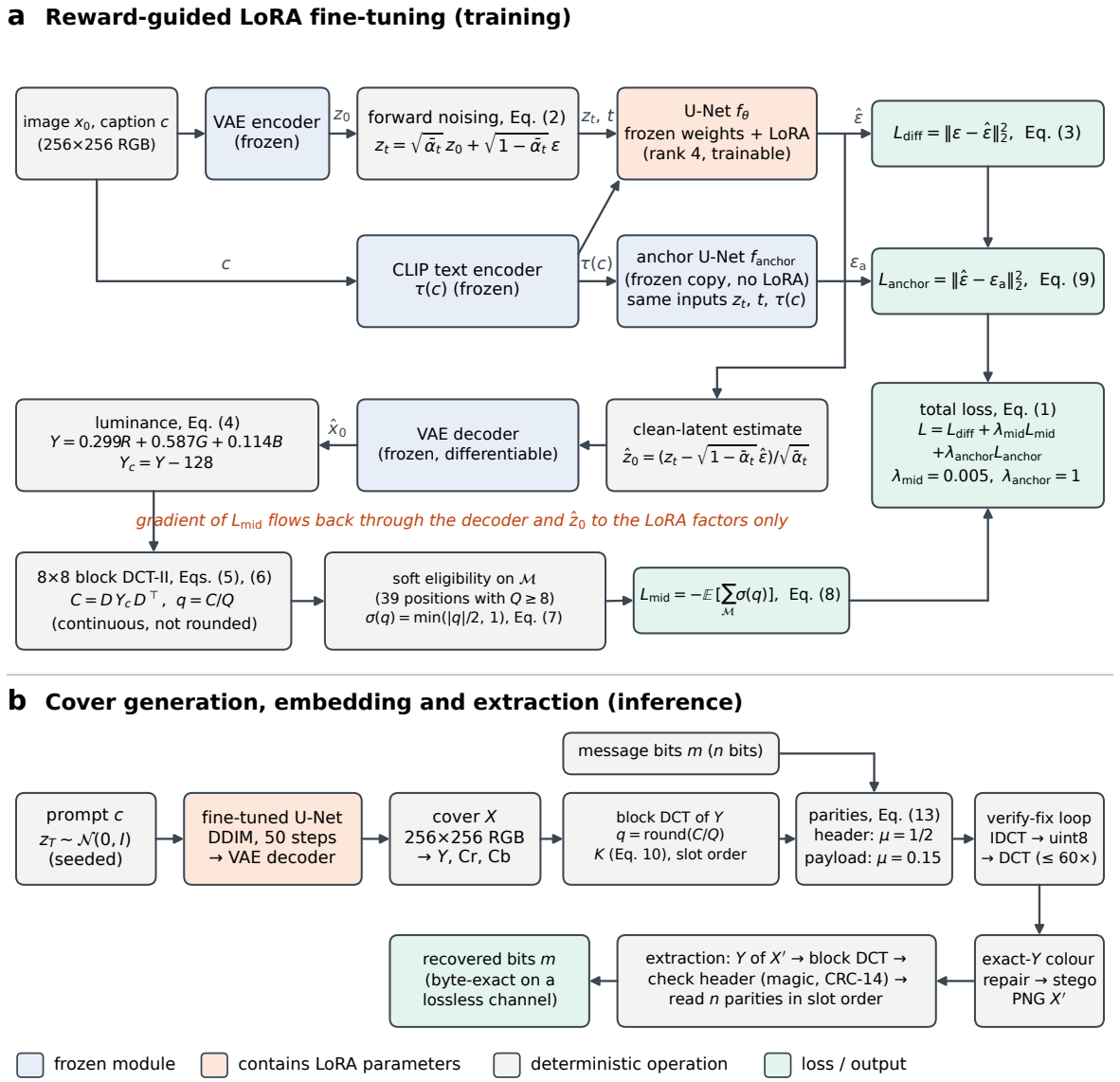


Figure 1. Overview of the DEDS framework. a, Reward-guided fine-tuning. A training image is encoded by the frozen variational autoencoder (VAE), noised, and denoised by a U-Net whose attention projections carry trainable low-rank (LoRA) factors. The predicted noise gives the diffusion loss and, through a one-step estimate of the clean latent decoded by the frozen VAE decoder, a differentiable reward on the block DCT of the luminance channel. A frozen copy of the U-Net provides the anchor loss. Only the LoRA factors receive gradients. **b**, Inference. The fine-tuned model generates a cover; the embedder computes the block DCT of the luminance channel, selects the number of coefficient positions per block, writes a 64-bit header and the payload into coefficient parities, repairs rounding errors, and stores the stego image losslessly. Extraction needs only the stego image. Symbols and equation numbers follow the Methods.

Characteristics of the generated covers

Fig. 3 shows five pairs from the corpus, each a baseline and a fine-tuned cover generated from the same prompt and seed, with the stego image of the fine-tuned cover. The fine-tuned covers are dense, fine-grained textures and differ visibly from their baseline counterparts, while each stego image is visually close to its own cover. Supplementary Fig. S1 shows five pairs chosen by a fixed rule, which include harsher, high-contrast textures. The mean CLIP [27] image-embedding cosine similarity between a baseline cover and its fine-tuned counterpart is 0.830 (95% confidence

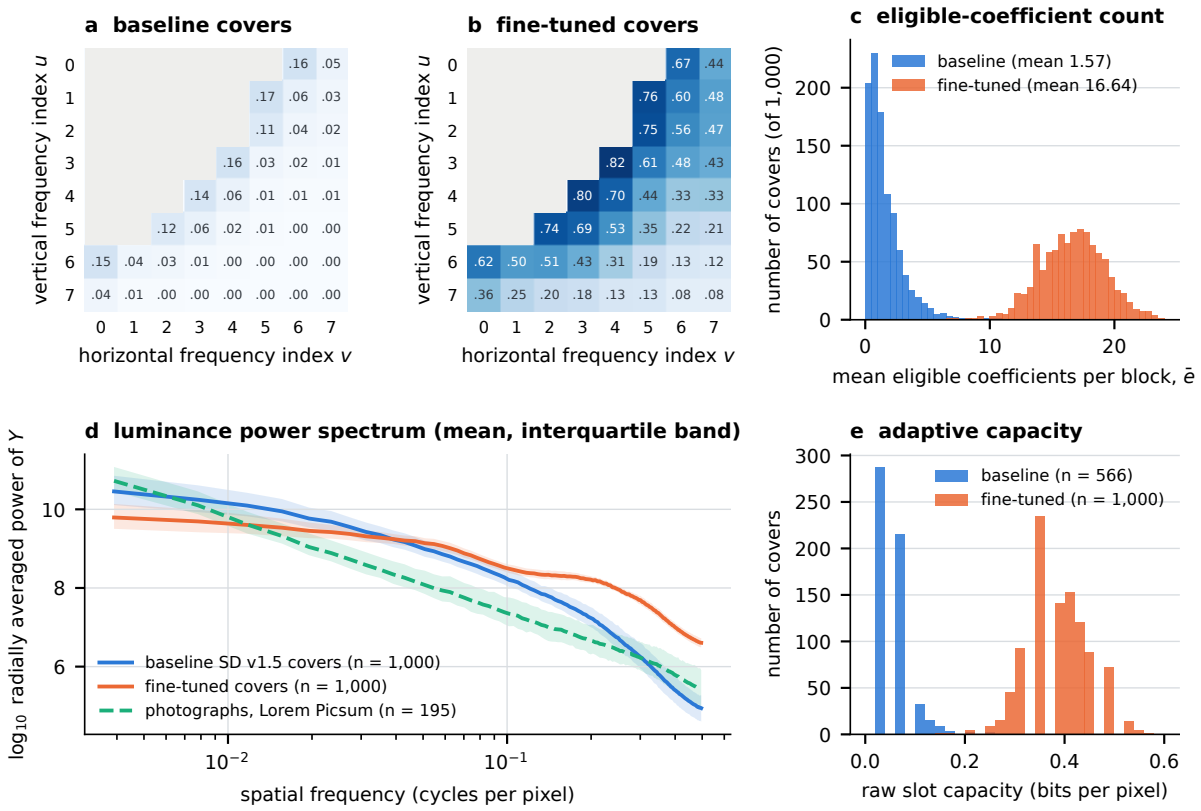


Figure 2. Effect of fine-tuning on DCT statistics and spectra. **a**,**b**, Fraction of 8×8 luminance blocks in which the quantised coefficient at position (u, v) has magnitude ≥ 2, averaged over 1,000 baseline (**a**) and 1,000 fine-tuned (**b**) covers. Numbers are fractions; grey cells lie outside the set $\mathcal{M}$ used for the reward and for embedding. **c**, Distribution of the mean eligible count per block, $\bar{e}$. **d**, Radially averaged power spectrum of the luminance channel (Hann window; line, mean of $\log_{10}$ power; band, interquartile range) for baseline covers, fine-tuned covers and 195 photographs from the Lorem Picsum service resized to 256×256. **e**, Raw slot capacity at each cover's own adaptive K; the 434 baseline covers that fall back to the fixed $K = 24$ are omitted.

interval 0.827 to 0.833), and the Fréchet inception distance (FID) [28] between the two sets is 238; with 1,000 images per set this FID is a coarse indicator only. By the same measures, embedding changes a fine-tuned cover very little (CLIP similarity 0.9998, FID 0.84; 0.9993 and 3.6 with bin-centre embedding).

The luminance power spectra show the change (Fig. 2d). Between 0.02 and 0.39 cycles per pixel the median log-log spectral slope is -3.68 for baseline covers, -1.88 for fine-tuned covers and -2.45 for a reference set of 195 photographs. Above 0.25 cycles per pixel lies a median of 16.3% of the luminance power of a fine-tuned cover, against 2.9% for a photograph and 0.8% for a baseline cover. The baseline covers are therefore smoother than photographs, and the fine-tuned covers carry about six times as much high frequency power as photographs. The reference photographs differ from the generated images in content and were JPEG-compressed at source, so this comparison indicates only the direction and rough size of the departure. Both generated sets were sampled without classifier-free guidance at half the native resolution of the model, which also affects their realism (Methods). With classifier-free guidance (scale 7.5) at the same resolution, 200 pairs gave 2.98 eligible coefficients per block for baseline and 16.22 for fine-tuned covers, so the gain does not depend on unguided sampling. At the native 512×512 resolution, which the adapter never saw in training, the fine-tuned covers still reached 9.88, more than twice the 4.48 of the baseline (Supplementary Table S15).

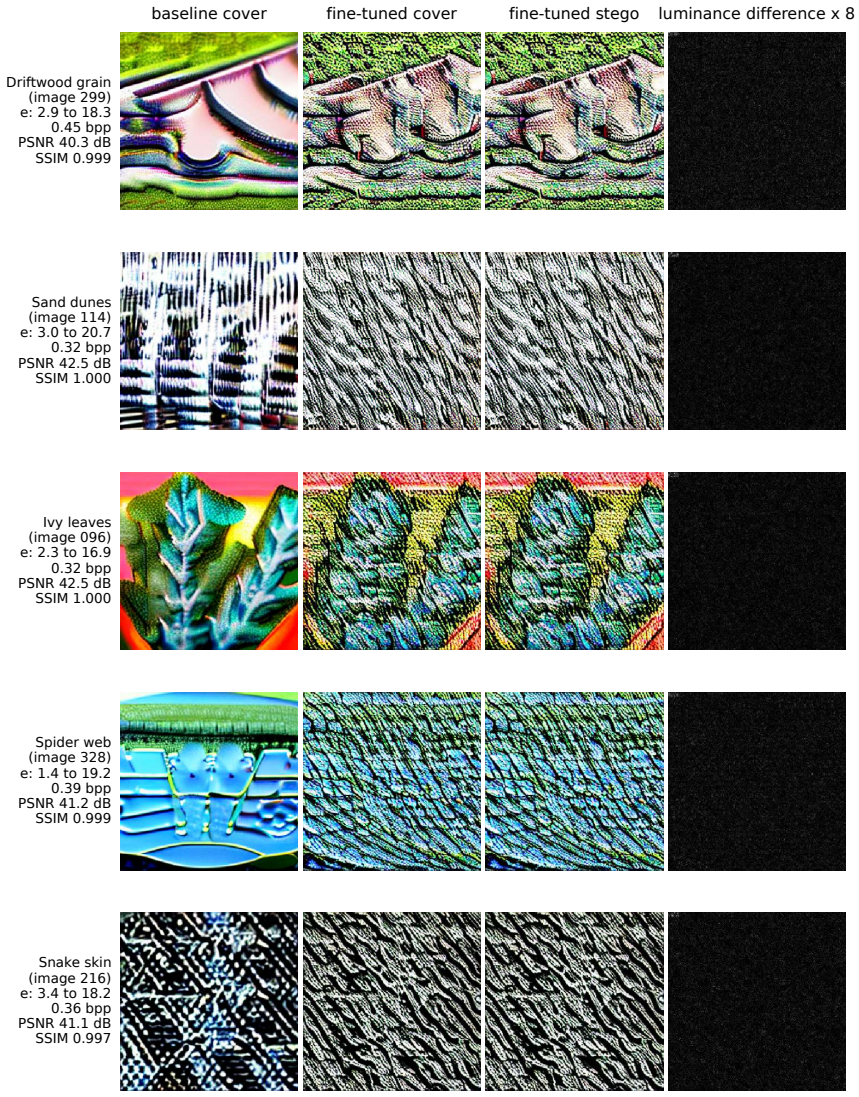


Figure 3. Example covers and stego images from the evaluation corpus. Five of the 1,000 pairs, selected by the authors as clear examples of five prompt subjects (driftwood grain, sand dunes, ivy leaves, spider web, snake skin). Columns: baseline cover; fine-tuned cover generated from the same prompt and seed; stego image of the fine-tuned cover; absolute luminance difference between stego and cover, amplified eight times. Row labels give the image index in the released corpus, the eligible count $\bar{e}$ of the baseline and fine-tuned cover, the embedded payload, and the luminance PSNR and SSIM of the fine-tuned stego image (margin-projected embedder, $\mu = 0.15$; with bin-centre embedding the same five images had 33.2 to 35.9 dB). All panels are the original 256×256 outputs of the stated experiments. Pairs selected by a fixed rule are shown in Supplementary Fig. S1.

Embedding capacity

Two quantities are kept separate. *Raw slot capacity* is the number of coefficient positions the embedder makes available, $1022\,(K + 1)$ bits for a 256×256 image. *Embedded payload* is the number of message bits actually written, n; bits per pixel (bpp) is $n/65536$ and excludes the 64-bit header. The test messages are random strings over a 67-character ASCII alphabet (about 6.07 bits of entropy per byte). With uniformly random bits instead of text, every message was still recovered exactly, and on 100 pairs the luminance PSNR advantage of fine-tuned over baseline covers rose from 1.42 to 1.87 dB (Supplementary Note 2).

At each cover's own adaptive K, fine-tuned covers offer a mean raw slot capacity of 0.389 bpp (range 0.156 to 0.577), and 924 of 1,000 exceed 0.30 bpp (Table 1; Fig. 2e). The baseline mean of 0.201 bpp overstates what those covers support: 434 of them average fewer than one eligible coefficient per block, and the embedder then substitutes a fixed $K = 24$ that does not reflect the cover. On the other 566 pairs the baseline covers offer 0.056 bpp and the fine-tuned covers 0.418 bpp, higher in every pair, a sevenfold gain. The adaptive rule only sets a slot budget and does not filter slots for eligibility; of the slots actually written at matched payload, 51.9% held an eligible coefficient on fine-tuned covers against 5.6% on baseline covers. Counting only slots that hold an eligible coefficient, fine-tuned covers offer 0.216 bpp against 0.011 bpp (Table 1).

To compare the two cover types at equal payload, each pair was assigned one target payload drawn uniformly from 0.30 to 0.60 bpp, and the same K and the same message were forced on both covers. Achieved payloads ranged from 0.300 to 0.600 bpp (mean 0.447; 19,680 to 39,304 bits; K from 19 to 38). For 688 of the 1,000 fine-tuned covers the forced K exceeded the cover's own adaptive K.

Exact recovery on a lossless channel

Reliability is the bit error rate (BER) and the exact byte-for-byte match rate after a save-to-PNG, reload and extract round trip. Every cover was embedded with a seed-reproducible message at its assigned payload: 1,000 of 1,000 fine-tuned and 1,000 of 1,000 baseline stego images returned the message byte-for-byte after the PNG round trip (BER = 0), with no embedding failure. The verify-fix loop needed a median of 15 passes (maximum 27) on fine-tuned covers and 14 (maximum 25) on baseline covers, well inside the limit of 60. Lossless formats other than PNG were not tested.

Embedding distortion at matched payload

Imperceptibility is measured using peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) [29] on the Y (luminance) channel, between each cover and its own stego image. Table 2 bins the 1,000 pairs by payload and Fig. 4 shows the individual images.

With the margin-projected embedder ($\mu = 0.15$), luminance PSNR over all pairs was 39.93 dB for fine-tuned and 38.31 dB for baseline covers, and the fine-tuned covers were better in every payload bin, by +1.31 dB at 0.30 to 0.35 bpp rising to +2.05 dB at 0.55 to 0.60 bpp (all $p < 10^{-21}$, rank-biserial $r \geq 0.87$; higher on the fine-tuned cover in 897 of 1,000 pairs). Luminance PSNR lay between 36.5 and 43.2 dB for fine-tuned covers. SSIM was 0.996 ± 0.005 (range 0.957 to 1.000) for fine-tuned against 0.948 ± 0.044 for baseline covers and was higher on the fine-tuned cover in all 1,000 pairs, with bin means 0.029 to 0.073 higher (all $p < 10^{-25}$). On all three colour channels PSNR was 38.06 ± 1.95 dB for fine-tuned and 36.17 ± 3.17 dB for baseline covers (mean $\pm$ s.d.).

The write rule accounts for most of the improvement. On the same pairs, payloads and K, the bin-centre rule gave 33.12 dB on fine-tuned and 33.03 dB on baseline covers, so the margin projection added +6.81 dB on fine-tuned covers (between +5.6 and +8.0 dB in every pair) and +5.28 dB on baseline covers (Table 2). Against plain bin-centre writing at the first write, the gain on fine-tuned covers matches the 5.61 dB of Eq. (14) to within 0.01 dB. The 6.81 dB is larger because the reference rule also applied the $|q'| \geq 2$ preference, which moves coefficients further, and because it compares the final images. For baseline covers the first-write gain is smaller because 94% of their slots hold $|q| \leq 1$, outside the model's uniform-offset assumption (Supplementary Note 3). Under the bin-centre rule the fine-tuned covers had lost their PSNR advantage above 0.50 bpp; with margin projection the advantage grows with payload, because their coefficients often already carry the right parity and are left alone.

Table 2. Luminance distortion at matched payload, binned by assigned payload ($n = 1{,}000$ pairs), margin-projected embedder ($\mu = 0.15$). Each pair was embedded with the same message and the same K on the baseline and the fine-tuned cover. Δ, fine-tuned minus baseline. p, two-sided paired Wilcoxon signed-rank test within the bin. r, matched-pairs rank-biserial correlation. For SSIM, $r = 1.00$ in every bin. The last three columns compare with the bin-centre rule ($\mu = \frac{1}{2}$ with the $|q'| \geq 2$ preference, our reference implementation) on the same pairs, payloads and K: PSNR of the fine-tuned stego images under that rule, and the gain of the margin-projected rule on fine-tuned (FT) and baseline (B) covers.

		PSNR_Y (dB)				SSIM_Y				vs bin-centre rule (dB)		
Payload (bpp)	n	Base	FT	Δ	p (r)	Base	FT	Δ	p	FT, centre	Gain FT	Gain B
0.30 to 0.35	177	41.05	42.36	+1.31	5.9×10^{-25} (+0.89)	0.970	0.998	+0.029	8.5×10^{-31}	35.86	+6.50	+5.72
0.35 to 0.40	153	39.72	41.16	+1.44	3.2×10^{-22} (+0.90)	0.962	0.997	+0.036	7.4×10^{-27}	34.50	+6.65	+5.53
0.40 to 0.45	180	38.69	40.17	+1.48	5.2×10^{-24} (+0.87)	0.953	0.997	+0.044	2.7×10^{-31}	33.36	+6.81	+5.49
0.45 to 0.50	160	37.76	39.48	+1.72	4.1×10^{-26} (+0.96)	0.946	0.996	+0.050	5.2×10^{-28}	32.62	+6.86	+5.21
0.50 to 0.55	181	36.73	38.49	+1.75	6.4×10^{-26} (+0.90)	0.935	0.995	+0.060	1.9×10^{-31}	31.48	+7.01	+5.05
0.55 to 0.60	149	35.67	37.72	+2.05	5.2×10^{-24} (+0.95)	0.921	0.994	+0.073	3.4×10^{-26}	30.65	+7.06	+4.63
All	1,000	38.31	39.93	+1.61	1.4×10^{-138} (+0.91)	0.948	0.996	+0.048	3.3×10^{-165}	33.12	+6.81	+5.28

Real photographs and simply processed covers

Could the same capacity be had without training, from real photographs or from baseline covers that are only sharpened or given added noise? We measured the 195 reference photographs, and the 1,000 baseline covers after an unsharp mask of four strengths or after Gaussian noise of three strengths, with the embedder's own eligibility rule (Table 3; Supplementary Table S12). None came close. The photographs had 1.94 eligible coefficients per block, and none of them could carry 0.30 bpp. Noise barely changed the count of the baseline covers, which rose from 1.57 to at most 1.92. Sharpening raised it further, to 7.69 at the strongest setting, which is still less than half of the 16.64 of the fine-tuned covers. Only 116 of the 1,000 most strongly sharpened covers reached 0.30 bpp, and the sharpening alone had already brought them down to 20.1 dB against the original cover.

We then embedded the same message with the same forced K in 200 pairs (every fifth pair). Fine-tuned stego images had a luminance PSNR 1.58 dB higher than baseline ones and 1.22 dB higher than the most strongly sharpened ones. The 195 photographs, given the K and message of pairs 0 to 194, scored 2.82 dB lower than the fine-tuned covers of those pairs (all $p < 10^{-29}$, paired Wilcoxon test). Covers with noise of $\sigma = 8$ were the one exception, 0.13 dB better than the fine-tuned covers ($p = 0.014$). By the embedder's rule, however, they offer only 0.057 bpp, so this payload had to be forced on them, and the noise itself had already lowered them to 34.3 dB against the original. Texture photographs also fell well short: the 5,640 images of the Describable Textures Dataset [30] averaged 2.74 eligible coefficients per block, and only 105 could carry 0.30 bpp (Supplementary Table S17).

Detectability

Detectability was measured with a detector trained for each embedder on stego images of that embedder (Table 4; Fig. 4c,d; Methods). The detectors were trained on 4,000 covers from the same two generators with seeds the corpus does not use, and tested on the 1,000 fine-tuned covers of the corpus with the DEDS stego images of Table 2, and with S-UNIWARD [8], WOW [7], HILL [31] and MiPOD [32] at 0.4 bpp. P_{E} and the area under the curve (AUC) are defined in Table 4.

The convolutional detector is SRNet [12] behind a fixed high-pass filter [33, 34]. For the four content-adaptive schemes it did not learn on fine-tuned covers: already on held-out fine-tuned covers from its own training distribution its output was almost constant and its AUC 0.50, against

Table 3. Fine-tuned covers compared with real photographs and with processed baseline covers. $\bar{e}$, mean eligible coefficients per block. ≥ 0.30 bpp, covers able to carry 0.30 bpp by the embedder's own rule (Eq. 10). Cover change: luminance PSNR of the processed cover against the original baseline cover. PSNR_Y: luminance PSNR of the stego image against its own cover at matched payload, with the same forced K and message, 200 pairs (every fifth); the photographs use the K and message of pairs 0 to 194, where the fine-tuned covers average 40.04 dB. All settings are in Supplementary Table S12.

Covers	$\bar{e}$	≥ 0.30 bpp	Cover change (dB)	PSNR_Y (dB)
Fine-tuned	16.64	924 of 1,000	none	39.89
Baseline	1.57	0 of 1,000	none	38.32
Real photographs	1.94	0 of 195	none	37.21
Baseline, unsharp mask, amount 4	7.69	116 of 1,000	20.1	38.68
Baseline, Gaussian noise, $\sigma = 8$	1.92	0 of 1,000	34.3	40.02

0.65 to 0.78 on baseline covers, so its chance-level results on the fine-tuned corpus covers reflect a detector that does not work on this cover source, not a property that hides the embedding. The DEDS detector was trained on stego images written with a margin of 0.25 at 0.4 bpp of random bits. With those settings it detected DEDS in the 1,000 fine-tuned corpus covers almost perfectly (AUC 0.997); on the stego images of Table 2, written with a margin of 0.15 at 0.30 to 0.60 bpp, its AUC was 0.710, which measures transfer to different embedding settings rather than an effect of the covers. We therefore base conclusions about detectability on the rich model.

The spatial rich model [10] with an ensemble of Fisher linear discriminants [11] detected all five spatial schemes on the same covers. Its AUC was 0.997 for DEDS, 0.973 to 0.989 for S-UNIWARD, WOW and HILL, and 0.851 for MiPOD, which was the hardest to detect. At the best threshold DEDS reached $P_{\mathrm{E}} = 0.015$, against 0.047 to 0.096 for S-UNIWARD, WOW and HILL. The threshold learned at a fixed 0.4 bpp did not transfer fully to the varying payloads of the corpus ($P_{\mathrm{E}} = 0.323$ for DEDS at the validation threshold), but the scores themselves separate DEDS stego images from covers almost completely. DEDS was tested at a higher average payload than the other schemes (0.447 against 0.4 bpp), so this comparison is approximate.

As a JPEG-domain reference, J-UNIWARD [8] and J-MiPOD [35] were embedded at 0.4 bits per non-zero AC coefficient in the same covers compressed at quality 75 and detected with DCTR features [36] and the same ensemble; their AUCs were 0.549 and 0.558. Detectors trained on photographs transfer poorly to a different cover source [37–39], which is why every detector here was trained on covers from the same generators as the corpus.

Message recovery after lossy processing

The main embedder is built for a lossless channel and is not meant to survive image processing (Supplementary Tables S1 and S13). On fine-tuned covers, requantisation with the embedder's own table kept 99.9, 93.5 and 78.1% of the payload bits at scales 1.0, 0.2 and 0.6, JPEG at quality 95 and 90 kept about 71%, and Gaussian noise of $\sigma = 2$ and 5 kept 85.0 and 71.1%; stronger JPEG compression, resizing, blur, cropping, shifts and rotation left the slots at chance. Because one wrong bit spoils a byte-for-byte match, no message survived any of these operations exactly (on baseline covers, 55 of 1,000 survived requantisation at scale 1.0). Eq. (15) protects a bit only until the image is rounded back to pixels, so at scale 0.2, where requantisation leaves only $0.05Q$ of the margin, 93.5% rather than all of the bits survive. The margin-projected rule keeps fewer bits after requantisation than the bin-centre rule, which kept 95 to 99% in a pilot on 17 covers; this is the price of the lower distortion.

An exploratory robust mode (Methods) writes a framed message repeatedly over all slots in a keyed order and decodes it by a vote weighted by each read's distance from the nearest bin edge.

Table 4. Detectability on the 1,000 fine-tuned covers of the corpus, with a detector trained for each embedder. Each detector was trained on 4,000 generated covers and their stego images from the embedder in the same row (0.4 bpp of random bits; 3,195 training and 399 validation pairs; for DEDS at a margin of 0.25, Methods) and tested on the 1,000 fine-tuned covers and their stego images. DEDS test images are the stego images of Table 2 (0.30 to 0.60 bpp); the other spatial schemes were embedded at 0.4 bpp; J-UNIWARD and J-MiPOD at 0.4 bits per non-zero AC coefficient (bpnzac) in the covers compressed at JPEG quality 75. P_{E}, mean of the false-alarm and missed-detection rates (lower means more detectable; 0.5 is chance), at the threshold fixed on the validation pairs (val.) and at the best threshold on the test scores (best). AUC, area under the ROC curve with 95% bootstrap interval over image pairs. SRNet-KV, SRNet with a fixed KV high-pass input filter; SRM+FLD, spatial rich model with the ensemble of Fisher linear discriminants; DCTR+FLD, DCTR features with the same ensemble.

Embedder	Payload (bpp)	Detector	P_{E} (val.)	P_{E} (best)	AUC (95% CI)
DEDS	0.30 to 0.60	SRNet-KV	0.465	0.329	0.710 (0.696 to 0.723)
S-UNIWARD	0.4	SRNet-KV	0.500	0.497	0.500 (0.499 to 0.500)
WOW	0.4	SRNet-KV	0.500	0.495	0.501 (0.500 to 0.502)
HILL	0.4	SRNet-KV	0.500	0.496	0.501 (0.500 to 0.501)
MiPOD	0.4	SRNet-KV	0.496	0.492	0.506 (0.505 to 0.507)
DEDS	0.30 to 0.60	SRM+FLD	0.323	0.015	0.997 (0.996 to 0.999)
S-UNIWARD	0.4	SRM+FLD	0.148	0.066	0.985 (0.981 to 0.988)
WOW	0.4	SRM+FLD	0.168	0.096	0.973 (0.966 to 0.979)
HILL	0.4	SRM+FLD	0.053	0.047	0.989 (0.986 to 0.992)
MiPOD	0.4	SRM+FLD	0.294	0.236	0.851 (0.831 to 0.870)
J-UNIWARD	0.4 bpnzac	DCTR+FLD	0.464	0.458	0.549 (0.541 to 0.558)
J-MiPOD	0.4 bpnzac	DCTR+FLD	0.457	0.450	0.558 (0.550 to 0.566)

On 200 fine-tuned covers (Supplementary Table S6), every message was recovered exactly at 0.005 and 0.02 bpp after own-table requantisation, JPEG at quality 95 and 90 and Gaussian noise of $\sigma = 2$ and 5. At 0.30 to 0.60 bpp it recovered 83 to 100% of the message bits under the same operations, mostly without exact recovery, and the weighted vote recovered 218 messages exactly where a plain majority vote recovered 188 (Supplementary Note 3). Stronger compression, blur, resizing and cropping stayed at chance, and robust mode costs image quality (about 33.1 dB), so it shows what redundancy buys on mild channels rather than making the method robust.

Loss weights and training length

Most of the change in DCT statistics happens early in training. In covers made from checkpoints saved every 5,000 steps, the mean eligible count of four validation covers rose from 2.67 per block for the baseline to 14.60 at step 5,000, then stayed between 14.07 and 16.84 until step 80,000 (Supplementary Table S2). All results here use the final checkpoint.

We also trained every combination of $\lambda_{\mathrm{mid}} \in \{0, 0.001, 0.005, 0.01, 0.05, 0.1\}$ and $\lambda_{\mathrm{anchor}} \in \{0, 0.1, 0.5, 1, 2\}$ for 5,000 steps and scored it by the average K of ten covers and by eye (Supplementary Table S3). A larger λ_{mid} raised K and a larger $\lambda_{\mathrm{anchor}}$ lowered it. From $\lambda_{\mathrm{mid}} \geq 0.05$, average K reached about 30 to 37, but the images were clearly unrealistic. We chose $\lambda_{\mathrm{mid}} = 0.005$ and $\lambda_{\mathrm{anchor}} = 1$ ($K = 22.3$) as a compromise between K and realism; distortion, spectra and detectability were not measured for each setting. In a 5,000-step ablation (Supplementary Table S18 and Fig. S3), the full objective gave 17.94 eligible coefficients per block, the anchor alone 2.08 and ordinary fine-tuning 1.80, so the gain comes mainly from the reward.

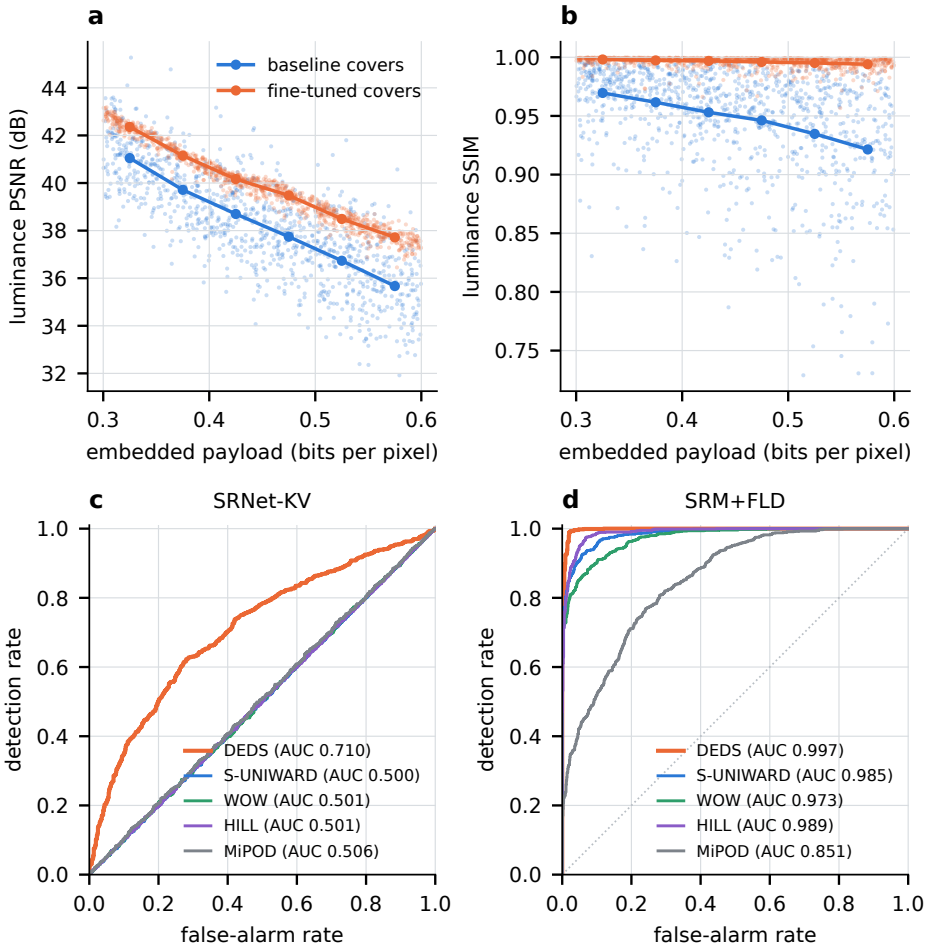


Figure 4. Embedding distortion and detectability. Luminance PSNR (**a**) and SSIM (**b**) of each stego image against its cover as a function of embedded payload, for baseline and fine-tuned covers embedded with the same message and the same K by the margin-projected embedder ($\mu = 0.15$; $n = 1{,}000$ pairs; points, individual images; lines, means in 0.05 bpp bins). ROC curves of the SRNet-KV (**c**) and SRM+FLD (**d**) detectors trained for each embedder (Table 4) and tested on the 1,000 fine-tuned covers and their stego images; the dotted line marks chance.

Computational cost

On an NVIDIA GeForce RTX 5060 (8 GB), generating one 256×256 cover with 50 DDIM steps took 1.9 s with the baseline model and 3.0 s with the fine-tuned model (adapter not merged), with peak GPU memory of 2.2 and 3.9 GB. With the block transform and the colour search vectorised, and three worker processes running alongside other processes on an AMD Ryzen 5 7500F processor, embedding a fine-tuned cover took a median of 0.7 s at 0.30 to 0.40 bpp and 1.5 s at 0.50 to 0.60 bpp (maximum 51 s), and extraction 0.02 s (Supplementary Table S4). Fine-tuning ran for 80,000 optimiser steps and took about 18 to 24 hours on an NVIDIA RTX 4080 (16 GB) by the authors' records. S-UNIWARD and WOW need no training or cover generation, so DEDS costs more per image.

Discussion

The experiments answer the question posed in the Introduction in two parts. First, a generator can be trained to supply the coefficients a quantised-DCT embedder needs. A rank-4 adapter trained with a differentiable block-DCT reward raised the mean eligible count from 1.6 to 16.6 coefficients per block on every one of 1,000 matched pairs and made it far more uniform across prompts. No baseline Stable Diffusion cover at this resolution had enough eligible coefficients of its own to support 0.30 bpp; nine in ten fine-tuned covers had. Real photographs did no better than the baseline, and sharpening or adding noise to the baseline covers stayed far below the fine-tuned covers while damaging the image, so the gain does not come for free from simple image processing. Second, the embedder recovers every message exactly after a PNG round trip across 0.30 to 0.60 bpp, at a luminance PSNR of 36.5 to 43.2 dB.

Supplementary Table S14 sets DEDS next to the closest generative and DCT domain methods. CRoSS [13] hides a secret image and targets degraded channels, GSD [14] and PSyDUCK [17] write the message into the generation process, so the receiver needs the generative model, and BAFLE-DCT [20] hides data in existing images. In DEDS the generator only makes the cover, and the stego image is decoded without the model or a key. Because these methods solve different tasks, we compared DEDS with established adaptive embedders on the same covers and with the same detectors instead (Table 4).

The write rule turned out to matter as much as the cover. Bin-centre writing moves every used coefficient, including the half whose parity is already correct, so its cost hardly depends on the cover; with it the fine-tuned covers improved SSIM but gained little PSNR. Projecting onto a shrunken bin leaves those coefficients alone, and the fine-tuned covers then gain 1.3 to 2.1 dB over baseline covers in every bin, while the corpus as a whole gains 6.8 dB over bin-centre writing. Eq. (14) predicts the first-write gain on fine-tuned covers to within 0.01 dB.

The gain in capacity comes with a change in the covers. The reward acts on the upper-frequency half of the DCT block, and the fine-tuned images carry about six times as much high frequency power as photographs. The anchor loss did not preserve the appearance of the generated images (CLIP similarity 0.83 to the baseline output of the same prompt and seed), and the covers are dense, repetitive textures. The generation settings contribute to this (no classifier-free guidance, half the native resolution, texture prompts only), and the present data cannot separate their effect from that of the reward. Whether a reward restricted to genuinely mid-frequency positions, a weaker reward, or guided sampling at native resolution yields usable capacity with natural spectra is the most important open question raised by this work.

Rich-model features detected every spatial scheme on the fine-tuned covers, and DEDS slightly more reliably than S-UNIWARD, WOW and HILL. This is expected for an embedder that writes parity into quantised coefficients without distortion-minimising coding. Our deep detector did not learn on the fine-tuned covers for the content-adaptive schemes, so it does not show whether such covers make embedding harder to find; a detector trained on a larger generated corpus would be needed to test that. Extraction is keyless, with a fixed identifier in the header; this keeps the receiver simple for a study of capacity and quality, and a keyed header could be added without changing the embedding.

The main mode is built for lossless storage or transmission, such as PNG files exchanged without recompression. DCT embedding of this kind needs the same quantisation table and block alignment at decode, so re-encoding breaks exact extraction, although mild operations left 71 to 99.9% of the payload bits intact. The redundancy mode recovered messages exactly after mild compression and noise at low payload. Operations that leave the reads at chance, such as resizing, cropping and blur, would need resynchronisation and embedding in coarser structures.

The corpus contains 1,000 pairs from one prompt bank of 100 texture subjects at one resolution, with one fine-tuned model and one random seed per prompt; 200 pairs from 20 other subjects and 300 pairs from six further domains, including paintings and cartoons, gave the same pattern (Supplementary Tables S7 and S16). The matched-pair design and the ablation (Supplementary Table S18) support a causal role for the reward, although the ablation used shorter training on less data than the main model. The steganalysis uses one network architecture and one rich-model feature set, trained on 4,000 generated covers and tested on 1,000 pairs, fewer than in standard benchmarks [40]; larger training sets would be expected to raise detection for all embedders. The detector training covers share prompts, but no prompt and seed pairs, with the corpus; on 200 pairs from prompts in neither set, the DEDS detector gave a similar AUC (0.720; Supplementary Table S7). Selection-channel-aware features such as maxSRM were not used, because the parity embedder does not define per-pixel change probabilities.

In summary, reward-guided fine-tuning of a diffusion model is an effective way to control the DCT statistics of generated covers and thereby the capacity of a quantised-DCT embedder. A margin-projected parity rule, whose distortion and tolerance we derived and checked numerically, turns that extra capacity into a clear quality gain: messages are recovered exactly on a lossless channel at 0.30 to 0.60 bpp with a luminance PSNR near 40 dB. The same experiments show that the capacity comes with atypical cover spectra and that robustness to lossy processing is not achieved. Rich-model features detect DEDS, so we do not claim security against steganalysis. Future work should constrain the reward so that generated spectra stay close to those of natural images, replace the fixed slot order with keyed, cost-guided coding so that changes go where they are hardest to detect (our implementation includes an experimental version), and embed in coarser, resynchronisable structures for lossy channels.

Methods

Training data

The training data came from `gmongaras/Imagenet21K_Recaption` on Hugging Face, a large captioned dataset built from ImageNet-21K [41]. Before fine-tuning, a filtering stage kept only textured images. For each image at its original resolution, the luminance (Y) channel was extracted using the BT.601 colour transform and split into non-overlapping 8×8 blocks. Each block was passed through a 2D Type II DCT and quantised with the luminance table Q (Supplementary Table S5). A block counted as eligible only if all of the first twenty AC positions in zigzag order (those with $1 \leq u + v \leq 5$) satisfied $|q_{u,v}| \geq 1$, where $q_{u,v} = \mathrm{round}(C_{u,v}/Q_{u,v})$. These low-frequency positions are a simple texture criterion and are distinct from the set $\mathcal{M}$ used for the reward and for embedding. An image was accepted if more than 30% of its blocks were eligible, a threshold set empirically to favour images with rich texture and exclude flat or low contrast ones. The filtering stopped once 50,000 images were accepted; accepted images were saved as JPEG at quality 95. They were then standardised: any image with a shorter side under 250 pixels was dropped, and the rest were centre cropped to their shorter side and resized to 256×256 using Lanczos resampling, leaving 40,101 training images. Each image's caption from the original dataset was kept for conditioning during fine-tuning, and pixel values were normalised to $[-1, 1]$. No duplicate or near-duplicate removal was performed. The dataset page states no licence; the images are subject to the ImageNet terms of access.

Model and LoRA adaptation

Stable Diffusion v1.5 is a latent diffusion model [24] that runs the denoising process in a compressed latent space. It consists of a variational autoencoder (VAE), a CLIP text encoder [27], a U-Net denoiser, and a noise scheduler. The VAE encoder compresses a $256 \times 256 \times 3$ image into a $32 \times 32 \times 4$ latent, and its decoder maps latents back to pixel space; because the decoder is differentiable, it provides the gradient path from the pixel-space reward back to the U-Net's LoRA parameters. During fine-tuning, the text encoder, the VAE, and all base U-Net weights stayed frozen, following the parameter efficient fine-tuning strategy of Hu et al. [25], which limits the weight update to a low rank subspace, $W = W_0 + (\alpha/r)BA$, with A and B trainable factors, rank r and scaling factor α; B is initialised to zero so training starts from the pre-trained checkpoint exactly. LoRA adapters of rank $r = 4$ and $\alpha = 8$ (dropout 0.05) were inserted into the query, key, and value projection matrices of every self attention and cross attention layer in the U-Net, giving 597,504 trainable parameters. No rank ablation was run. Base weights were stored in bfloat16, LoRA parameters in float32, with gradient checkpointing on the U-Net.

Training objective

The optimisation objective combines diffusion, reward, and anchor losses:

$$L = L_{\text{diff}} + \lambda_{\text{mid}}\, L_{\text{mid}} + \lambda_{\text{anchor}}\, L_{\text{anchor}}, \tag{1}$$

with $\lambda_{\text{mid}} = 0.005$ and $\lambda_{\text{anchor}} = 1.0$.

Diffusion loss. L_{diff} is the standard ε-prediction objective [42] used to train latent diffusion models. At each step, a clean image x_0 is encoded by the frozen VAE encoder into a latent z_0. A timestep t is sampled uniformly from the 1,000 training timesteps, and Gaussian noise $\varepsilon \sim \mathcal{N}(0, I)$ is added per the forward process:

$$z_t = \sqrt{\bar{\alpha}_t}\, z_0 + \sqrt{1 - \bar{\alpha}_t}\, \varepsilon, \tag{2}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t}(1 - \beta_i)$ is the cumulative product of noise schedule coefficients. The trainable U-Net f_θ predicts the added noise conditioned on the text embedding $\tau(c)$ of caption c:

$$L_{\text{diff}} = \|\varepsilon - f_\theta(z_t, t, \tau(c))\|_2^2 . \tag{3}$$

DCT reward loss. L_{mid} biases the fine-tuned model toward images whose Y channel block DCT coefficients are large at the positions the embedder uses. At each step, a pixel space estimate $\hat{x}_0$ is obtained by computing the predicted clean latent $\hat{z}_0 = (z_t - \sqrt{1 - \bar{\alpha}_t}\,\hat{\varepsilon})/\sqrt{\bar{\alpha}_t}$ from the current denoiser output $\hat{\varepsilon}$ and decoding it through the frozen VAE decoder. Since the decoder is a deterministic, differentiable function of the latent, gradients of L_{mid} propagate through the decoder back to the LoRA parameters. After rescaling $\hat{x}_0$ to $[0, 255]$, the BT.601 luminance channel is

$$Y = 0.299R + 0.587G + 0.114B. \tag{4}$$

The level shifted luminance $Y_c = Y - 128$ is divided into non-overlapping 8×8 blocks, and the 2D Type II DCT of each block is

$$C[k, l] = \sum_i \sum_j D[k, i]\; Y_c[i, j]\; D[l, j], \tag{5}$$

where D is the orthonormal DCT-II basis matrix with entries

$$D[k, n] = a_k \cos\left(\frac{\pi(2n + 1)k}{16}\right), \qquad a_0 = \sqrt{1/8}, \quad a_k = \sqrt{2/8} \;\; (k > 0). \tag{6}$$

Each coefficient is normalised without rounding, $q[k,l] = C[k,l]/Q[k,l]$, where Q is the quantisation table in Supplementary Table S5. A differentiable eligibility score is evaluated at each position of the set $\mathcal{M}$ (the 39 positions with $Q[k,l] \geq 8$) using a clipped linear function,

$$\sigma(q) = \min(|q|/2,\ 1)\,, \tag{7}$$

and the reward loss is the negative mean, over all blocks and images in the batch, of the per-block sum:

$$L_{\text{mid}} = -\,\mathbb{E}_{\text{blocks}}\left[\sum_{(k,l)\in\mathcal{M}} \sigma(q[k,l])\right]. \tag{8}$$

The threshold $Q \geq 8$ was chosen for stability of the parity under rounding; with the table used here it selects the upper-frequency half of the block (Fig. 2), and the subscript "mid" is retained only for continuity with the released code.

Anchor loss. A frozen copy of the original U-Net, the anchor network f_{anchor}, is maintained throughout training to limit drift of the trainable U-Net from the pre-trained model. The anchor loss is the mean squared error between the noise predictions of the trainable and frozen networks on the same noisy latent and conditioning:

$$L_{\text{anchor}} = \|f_\theta(z_t, t, \tau(c)) - f_{\text{anchor}}(z_t, t, \tau(c))\|_2^2\,. \tag{9}$$

Optimisation

LoRA parameters were optimised using AdamW [43] (learning rate 1×10^{-5}, weight decay 1×10^{-2}, $\beta_1 = 0.9$, $\beta_2 = 0.999$). A linear warm up schedule was applied over the first 1,000 steps, after which the learning rate was held constant. Gradient norms were clipped at 1.0, and gradients were accumulated over 4 micro-batches of 2 images (effective batch size 8). Training ran for 80,000 optimiser steps with checkpoints every 5,000. An exponential moving average of the LoRA parameters (decay 0.9999) was saved with each checkpoint but not evaluated: the covers here use the final-step optimiser weights, and regenerating the first twelve covers of each set from them with the released generation function reproduced the stored images pixel for pixel.

Cover generation

Covers were generated at 256×256 pixels with the DDIM sampler [44] (50 steps, deterministic) from a single conditional forward pass per step, that is, without classifier-free guidance. The prompt bank has 1,000 prompts from 100 texture subjects, each with the same ten modifiers. They were generated in two batches with the same code and weights, 400 prompts from 40 subjects and then 600 from 60 further subjects. Prompt i was sampled with seed $42 + i$ for both the baseline and the fine-tuned model, so that the two members of a pair share the prompt and the initial noise and differ only in the adapter weights. Decoded images were clipped to $[-1, 1]$, converted to 8-bit and stored as PNG. For the check with classifier-free guidance, prompts 0, 5, ..., 995 were sampled at 256×256 and prompts 0, 10, ..., 990 at 512×512 with guidance scale 7.5, the same seeds and the same sampler; corpus image 000 of both models, regenerated with the corpus settings by the same code, was identical to the stored cover.

Embedding

Embedding operates on the luminance channel obtained with the 8-bit integer BT.601 transform of OpenCV. The channel is level shifted, split into 8×8 blocks and transformed with the orthonormal

2D DCT; coefficients are quantised as $q[u,v] = \mathrm{round}(C[u,v]/Q[u,v])$. No JPEG file is produced at any stage; the quantisation table serves only to define parities.

Coefficient positions and adaptive capacity. The 39 positions of $\mathcal{M}$ are ordered by zigzag index, and the adaptive parameter K selects the first $K+1$ of them for use in every block. K is determined per image. For each non-header block, the system counts positions in $\mathcal{M}$ whose quantised magnitude satisfies $|q[u,v]| \geq T$ with $T = 2$. With $\bar{c}$ the mean count across blocks,

$$K = \min(38,\ \max(1,\ \lfloor \bar{c} \rfloor + \lfloor \lfloor \bar{c} \rfloor / 2 \rfloor)), \tag{10}$$

and a fixed fallback $K = 24$ is used when $\lfloor \bar{c} \rfloor = 0$. The raw slot capacity in bits is

$$\mathrm{capacity} = N_{\mathrm{payload}}\,(K+1), \tag{11}$$

where $N_{\mathrm{payload}} = 1022$ is the number of non-header blocks in a 256×256 image. Capacity counts every selected position in every block, whether or not the coefficient there is eligible. Payload slots (i,j,u,v) are filled in a fixed order: by increasing quantisation step $Q[u,v]$, then by zigzag rank, then by block raster index, so that positions with small steps are consumed first. This order depends only on K and the image size.

Header. A 64 bit header is embedded into two designated blocks, at the 32 positions of $\mathcal{M}$ with the largest Q in each, and written with the bin-centre rule ($\mu = \frac{1}{2}$ in Eq. (13)) because one wrong header bit loses the whole message. It encodes a 16 bit identifier (0x4447), a 24 bit payload bit count, a 6 bit encoding of K, a 4 bit version, and a 14 bit integrity check taken from the low 14 bits of the CRC-32 computed over the preceding 50 bits, zero padded to 56 bits.

Parity embedding with a margin. A bit b is carried by the parity of the quantised coefficient $q = \mathrm{round}(C/Q)$. Write the bin of an integer q' as $[(q' - \frac{1}{2})Q,\ (q' + \frac{1}{2})Q]$ and its admissible part, at a distance μQ from both edges, as

$$I_\mu(q') = \left[(q' - \tfrac{1}{2} + \mu)Q,\ (q' + \tfrac{1}{2} - \mu)Q\right], \qquad 0 < \mu \leq \tfrac{1}{2}. \tag{12}$$

With Π_I the projection (clipping) onto an interval I, the embedded coefficient is

$$C' = \begin{cases} \Pi_{I_\mu(q)}(C), & q \bmod 2 = b, \\ \arg\min\limits_{x \in \{\Pi_{I_\mu(q-1)}(C),\ \Pi_{I_\mu(q+1)}(C)\}} |x - C|, & q \bmod 2 \neq b. \end{cases} \tag{13}$$

A coefficient whose parity already matches is left where it is unless it lies within μQ of a bin edge; a coefficient whose parity must change crosses the nearer edge and stops μQ inside the neighbouring bin. At $\mu = \frac{1}{2}$ each admissible interval shrinks to its bin centre and Eq. (13) becomes the bin-centre rule, our reference implementation, which wrote every used coefficient at $q'Q$ and, when a parity had to change, preferred $|q'| \geq 2$ (that preference is not used here). We use $\mu = 0.15$, chosen in a pilot on 50 covers in which $\mu = 0.25$, 0.15 and the $|q'| \geq 2$ preference were compared with the bin-centre rule.

Expected distortion. Write the offset of a coefficient in its bin as $x = C - qQ$ and assume, following high-resolution quantisation theory, that x is uniformly distributed on the interval $[-Q/2, Q/2]$. We also assume that the message bits are independent of the cover and equally likely to be 0 or 1, so each parity case has a probability of $\frac{1}{2}$. The test messages of the main experiments are text, for which this holds only approximately; Supplementary Note 2 compares them with uniformly random bits. If the parity already matches, the coefficient moves only when $|x| > (\frac{1}{2} - \mu)Q$, which has probability 2μ, and then by $t = |x| - (\frac{1}{2} - \mu)Q$, uniform on $[0, \mu Q]$; the conditional mean square is $2\mu \cdot \mu^2 Q^2/3$. If the parity must change, let $s = Q/2 - |x|$ be the distance to the nearer edge, uniform on $[0, Q/2]$; the coefficient moves by $s + \mu Q$, so the

conditional mean square is

$$\frac{2}{Q}\int_0^{Q/2}(s+\mu Q)^2\,ds=\frac{2}{3}\Big[(\tfrac{1}{2}+\mu)^3-\mu^3\Big]Q^2.$$

Averaging the two cases,

$$D_\mu=E\big[(C'-C)^2\big]=\frac{(\frac{1}{2}+\mu)^3}{3}\,Q^2. \tag{14}$$

At $\mu=\frac{1}{2}$ this is $Q^2/3$, the cost of bin-centre writing ($Q^2/12$ for a matching parity, $7Q^2/12$ for a change). The gain over bin-centre writing is therefore $G(\mu)=-30\log_{10}(\frac{1}{2}+\mu)\,\mathrm{dB}$: 3.7 dB at $\mu=0.25$, 5.6 dB at $\mu=0.15$ and 9.0 dB as $\mu\to 0$. Because the DCT is orthonormal, Parseval's identity makes the same quantity the squared change of the pixel block before rounding.

Guaranteed tolerance. Every written coefficient lies in $I_\mu(q')$, so a later change e leaves its parity intact whenever

$$|e|<\mu Q. \tag{15}$$

Two consequences follow. First, rounding the inverse transform to integers adds an error of variance 1/12 to each pixel, and the errors of different pixels are close to independent; the rows of an orthonormal transform have unit norm, so each coefficient receives the same variance, a standard deviation of $1/\sqrt{12}\approx 0.29$. With $Q\geq 8$ on $\mathcal{M}$, $\mu=0.15$ leaves a margin of at least 1.2, about four standard deviations. Second, requantising with the embedder's table scaled by s moves a coefficient by at most half the scaled step, about $sQ/2$, so the written parities survive the requantisation whenever $\mu>s/2$. This holds before the result is rounded back to pixels; that rounding can still flip a parity when the margin left over, $(\mu-s/2)Q$, is small. With the unscaled table every coefficient returns to the centre of its own bin and no parity changes for any μ. Pixel clipping after the inverse transform is not covered by these bounds. The margin thus sets the balance between distortion, Eq. (14), and tolerance, Eq. (15), explicitly.

Numerical check. Supplementary Note 3 and Supplementary Fig. S2 test each result. On two million synthetic coefficients with uniform offsets, Eq. (14) is met to within 0.03% for every μ from 0.05 to 0.5. On the first write pass of 100 fine-tuned covers at the Table 2 payloads, the measured gain over bin-centre writing equals $G(\mu)$ to within 0.01 dB (5.61 dB at $\mu=0.15$); on baseline covers it is smaller (4.67 dB) because 94% of their slots hold $|q|\leq 1$, where the uniform-offset assumption fails. The rounding noise measured on the same images has a standard deviation of 0.288 against the predicted 0.289, Parseval's identity holds to 10^{-15}, and after scaled-table requantisation all parities survived in the coefficient domain wherever $\mu>s/2$.

Verify-fix loop and colour reconstruction. Since the inverse DCT and subsequent 8 bit rounding can alter quantised values, a verify-fix loop follows the initial embedding pass. The embedded Y channel is reconstructed via inverse DCT and rounded to 8 bits, blocks are re-transformed, and every bit position is re-verified against the intended payload. Rounding and, in saturated blocks, clipping at 0 and 255 can undo a small move, so a slot that reads wrongly for the f_k-th time is rewritten with Eq. (13) at the margin

$$\mu_k=\min\Big(\tfrac{1}{2},\ \mu\,2^{\,f_k-1}\Big). \tag{16}$$

A slot that keeps failing therefore reaches the bin centre, the most tolerant position, after at most $1+\lceil\log_2(1/2\mu)\rceil$ failures (three at $\mu=0.15$), while slots that never fail keep the small margin and its low distortion. A variant that also compensated for clipping losses failed to converge on two clipped baseline covers, so it was dropped and only the per-slot escalation was kept. The loop continues until no disagreement remains or 60 iterations have been made. The target Y channel is then merged with the unchanged Cr and Cb channels and converted to BGR. Because the 8-bit colour conversion is not exactly invertible, a selective repair step is applied to any 8×8

block in which a bit reads wrongly after conversion: for every pixel whose luminance disagrees with the target, the colour closest in L_2 distance that produces the target value under

$$Y = (4899R + 9617G + 1868B + 8192) \gg 14 \tag{17}$$

is found by search over a progressively widening neighbourhood. The vectorised search returns exactly the colour of a pixel-by-pixel search, ties included. The stego image is stored as PNG.

Failure conditions. Embedding stops with an error if the message exceeds the raw slot capacity or if the verify-fix loop has not converged after 60 iterations; neither occurred in the 2,000 embeddings at 0.30 to 0.60 bpp or in the 400 embeddings of the unseen-prompt check. For dimensions that are not multiples of 8, border rows and columns are skipped by the block transform and are not guaranteed identical to the cover; all images here are 256×256. The payload length field limits a message to $2^{24} - 1$ bits, and an empty message is not supported. In the main mode the integrity check covers the header only and the payload carries no checksum, so a stego image altered after embedding can return a wrong message without any warning; in 400 trials with the bin-centre implementation that each flipped one payload bit of a test stego image, 346 returned altered text with no error and 54 raised a text decoding error. The robust mode below adds a checksum over the message.

Extraction. The extractor converts the stego image to YCrCb, isolates the Y channel, and applies the same block DCT. The header is read from the two header blocks; if either the identifier or the check fails, extraction is aborted. Otherwise the extractor rebuilds the slot order from the recovered K and the image dimensions and reads the parities of the first n slots. No key, seed, or sidecar file is used. Supplementary Note 1 lists every step of embedding and extraction.

Payload. A message is the UTF-8 byte string written most significant bit first. In the embedding experiments the messages were random strings over a 67-character alphabet (letters, digits, space and four punctuation marks) and were embedded without compression. The exceptions are the check in Supplementary Note 2 and the training images of the steganalysis detectors, which used uniformly random bits. Payload in bits per pixel is

$$\text{bpp} = n/(H\,W), \tag{18}$$

with n the number of message bits and $H = W = 256$; the header is not counted. The released training pipeline contains an optional canonical Huffman codec, which was not used for the results in this paper.

Robust mode

To measure how far redundancy can go on lossy channels, the same embedder has an exploratory robust mode, signalled by the version field of the header. All 39 positions of $\mathcal{M}$ in every non-header block are used, $n_s = 39\,858$ slots, visited in a pseudorandom order fixed by a key (a public default key here). A message of m bytes is wrapped in a frame of $L = 8m + 32$ bits: a 16 bit byte count, the message and a 16 bit CRC (CCITT polynomial). Frame bit i is written into every slot k with $k \bmod L = i$, so each frame bit has $\lfloor n_s/L \rfloor$ or $\lceil n_s/L \rceil$ copies; the redundancy follows from the message length alone, about 110 copies at 0.005 bpp and one at 0.6 bpp. Because the slot order is a keyed permutation, the copies of a frame bit lie in unrelated blocks and frequency positions, which spreads damage that is local in the image or in frequency. Copies are written with Eq. (13) at $\mu = \frac{1}{2}$, the most tolerant setting.

Reliability-weighted vote. A slot is read as a parity b_k together with its offset $f_k = C_k/Q_k - \text{round}(C_k/Q_k) \in [-\frac{1}{2}, \frac{1}{2}]$, and we define its reliability

$$a_k = 1 - 2|f_k| \in [0, 1]. \tag{19}$$

Frame bit i is decided by the sign of

$$v_i = \sum_{k:\, k \bmod L = i} (2b_k - 1)\, a_k, \tag{20}$$

which reduces to a majority vote when every $a_k = 1$. The weight can be derived. A copy is written at a bin centre, and a channel adds a perturbation e with a density p that is symmetric and decreasing in $|e|$. An observed offset f is explained either by $e = fQ$, with the parity read correctly, or, in a nearest-bin approximation that considers only the nearest bin of the other parity, by a perturbation of size $(1 - |f|)Q$, with the parity read wrongly. The likelihood ratio of the two is $p(|f|Q)/p((1 - |f|)Q)$, which decreases as $|f|$ grows, so the probability that a read is correct increases with a. For Gaussian p with variance σ^2 the log-likelihood ratio is

$$\ln \frac{p(|f|Q)}{p\big((1 - |f|)Q\big)} = \frac{\big((1 - |f|)^2 - f^2\big)Q^2}{2\sigma^2} = \frac{Q^2}{2\sigma^2}\, a, \tag{21}$$

exactly proportional to a. The log-likelihood ratio of copy k is therefore $a_k Q_k^2/2\sigma_k^2$, and Eq. (20) drops the factor $Q_k^2/2\sigma_k^2$. Because the keyed order places the copies of a frame bit at different positions, whose steps and noise levels differ, the vote approximates the sum of per-copy log-likelihood ratios rather than equalling it; weighting each copy by Q_k^2/σ_k^2 would make it exact under this model, and we did not use that weighting. A candidate frame is accepted if the decoded byte count equals m and the CRC agrees. If the header is damaged, every byte count is tried in turn, testing the count field first and the CRC only when it agrees. Supplementary Note 3 shows that single-read accuracy rises steadily with a (for example from 54% to 94% under JPEG at quality 90) and compares the weighted vote with the majority vote on the same reads.

Matched-payload protocol

Each pair was assigned a target payload drawn uniformly from $[0.30, 0.60]$ bpp. The position parameter was set to $K = \lceil n_{\text{target}}/1022 \rceil - 1$ and forced on both covers of the pair, overriding Eq. (10), and the same message was embedded in both. PSNR and SSIM were computed with scikit-image between the luminance channels of cover and stego. All 2,000 stego images were produced in one run of the margin-projected embedder ($\mu = 0.15$); the message for pair i was drawn from a pseudorandom generator seeded with $424242 + i$ over the 67-character alphabet. Each stego image was written to PNG, reloaded and extracted, and the output compared byte-for-byte with the input. The bin-centre comparison used the same K, payload and message for every pair.

Steganalysis

Detector training data. The detectors were trained on 4,000 covers generated with the cover-generation procedure above, 2,000 from each model, with seeds from 100000 upwards, so that no prompt and seed pair of the evaluation corpus was reused. All detectors work on the luminance plane (256×256, 8 bits). Every training cover was embedded with 0.4 bpp of uniformly random bits by each embedder. The DEDS training stego images were produced with our embedder at a margin of $\mu = 0.25$ and 26,208 random bits per image (0.4 bpp), with K forced from the payload, and each was read back bit for bit; the corpus stego images used for testing were produced with $\mu = 0.15$ at 0.30 to 0.60 bpp, so the DEDS detectors were tested on embedding settings different from those they were trained on. Training covers were generated with prompts from the first 400 prompts of the bank, so prompts but no prompt and seed pairs are shared with the corpus; the 200 pairs of Supplementary Table S7 use prompts that appear in neither. S-UNIWARD, WOW, HILL and MiPOD were simulated at the entropy bound [45] with the conseal library

(version 2025.11), and the same simulation at 0.4 bpp was applied to the 1,000 fine-tuned and 1,000 baseline covers of the corpus for testing. The pairs were split into 3,195 training, 399 validation and 406 held-out pairs, with a cover and its stego image always in the same subset. The decision threshold minimising P_{E} on the validation pairs was fixed before any corpus image was scored. AUC intervals are percentile intervals from 2,000 bootstrap resamples of image pairs.

Convolutional detector. SRNet [12] was preceded by the fixed 5×5 KV high-pass filter [33,34], whose output is multiplied by 4. Without it, SRNet stayed at chance on the training pairs even at 1.0 bpp. Training used AdamW (learning rate 2×10^{-4} with cosine decay, weight decay 2×10^{-4}), 16 pairs per batch with a cover and its stego image in the same batch, the same random flip and 90° rotation for both images of a pair, and early stopping on validation P_{E} with a patience of 15 epochs (at most 80). For the content-adaptive schemes each detector was first trained for 30 epochs at 1.0 bpp and then fine-tuned at 0.4 bpp from those weights [46]; the DEDS detector learned at 0.4 bpp directly. Test scores were averaged over the eight flips and rotations of each image. On the 406 held-out pairs of their training task the detectors reached AUCs of 0.59 to 0.67 for the content-adaptive schemes, all of it from baseline covers (0.65 to 0.78; 0.50 on fine-tuned covers), and 0.993 for DEDS.

Rich-model and JPEG detectors. SRM features [10] (34,671 dimensions) and DCTR features [36] (8,000 dimensions) were extracted with the sealwatch library (version 2025.9) and classified with the ensemble of Fisher linear discriminants [11], with the number of base learners and the subspace dimension chosen automatically from the out-of-bag error. On the 406 held-out pairs the rich model reached AUCs of 0.97 to 0.99 for the content-adaptive schemes and 1.000 for DEDS. For the JPEG reference, the covers were compressed at quality 75 with jpeglib (libjpeg 6b); J-UNIWARD was simulated with conseal, and J-MiPOD was implemented on conseal's MiPOD components following its published description [35], both at 0.4 bits per non-zero AC coefficient.

Processing tests

Main mode. The 2,000 stego images of the matched-payload sweep were each processed by one of: requantisation of the luminance channel with the embedder's own table scaled by 1.0, 0.6 or 0.2 (the scale factors of JPEG qualities 50, 70 and 90; chrominance unchanged); JPEG compression with OpenCV at quality 95, 90, 70 or 50; bicubic downscaling by a factor of 0.90, 0.75 or 0.50 followed by bicubic upscaling; additive Gaussian noise with standard deviation 2, 5, 10 or 15 grey levels; Gaussian blur with a 3×3 kernel and $\sigma = 0.5$; removal of a 5% border followed by bicubic resizing; requantisation with a non-standard table; shifts by one and two pixels and a 1° rotation with border replication; and removal of a 2 pixel border followed by bilinear resizing. For each image we recorded the percentage of payload bits read correctly at the known slot positions, whether the header was intact, and whether the extractor returned the exact message.

Robust mode. Two hundred fine-tuned covers were drawn at random (seed 2026), and each was assigned at random one of eight net payloads, 0.005, 0.02, 0.05, 0.1, 0.2, 0.3, 0.45 and 0.6 bpp (25 covers each), with a message seeded by $777000 + i$. The same operations were applied. We recorded the percentage of message bits recovered after the vote of Eq. (20) with the frame length known, and whether the extractor returned the exact message with no side information. For the ASCII test messages a decoder that has lost all information still matches about 54% of the message bits, the share of zero bits, so only values clearly above that indicate recovered information.

Photographs and processed covers

The photographs are the 195 Lorem Picsum images of the spectral comparison, resized to 256×256. Sharpened covers were made from each baseline cover x, on every colour channel, as $x + a\,(x - G_1 * x)$, where $G_1 * x$ is x blurred with a Gaussian of standard deviation one pixel and the amount a is 0.5, 1, 2 or 4. Noisy covers were made by adding Gaussian noise of standard deviation 2, 4 or 8 grey levels with a fixed seed per image. Both were rounded and clipped to 8 bits. Eligible counts, K and capacity were computed as for the corpus. For the matched payload test, the strengths closest to the fine-tuned eligible count (amount 4 and $\sigma = 8$) were embedded with the K and message of the matched payload protocol and scored against their own covers.

Cover statistics and spectra

CLIP similarity used the image encoder of `openai/clip-vit-base-patch32`; FID used 2,048-dimensional Inception features. Radial power spectra were computed from the 2D Fourier transform of the mean-subtracted, Hann-windowed luminance channel and averaged over annuli of unit width. The log-log spectral slope was computed for each image from its radial spectrum between 0.02 and 0.39 cycles per pixel, and the median over images is reported. The share of power above 0.25 cycles per pixel is the power of all 2D frequencies beyond that radius divided by the total power without the DC term.

Further cover checks and ablation

Three further checks used the same embedder and matched-payload protocol: 300 pairs from six new domains, the 5,640 Describable Textures Dataset photographs [30] and a 5,000-step ablation of four adapters (Supplementary Note 4).

Statistical analysis

Baseline and fine-tuned values were compared with two-sided paired Wilcoxon signed-rank tests, on the whole corpus and within payload bins; bins were not pooled for the per-bin tests. Effect size is the matched-pairs rank-biserial correlation computed from the signed ranks. The twelve per-bin tests form one family; the reported p values are uncorrected and descriptive. The largest uncorrected p is 3.2×10^{-22} (PSNR at 0.35 to 0.40 bpp), so Holm correction does not change the conclusions. No pair had a zero difference. Because each of the 100 subjects contributes ten pairs, we repeated the main comparisons on subject means: luminance PSNR, SSIM and the eligible count were higher for fine-tuned covers in all 100 subjects ($p = 3.9 \times 10^{-18}$ for each).

Computational environment

Fine-tuning and the first 400 pairs ran on an NVIDIA RTX 4080 (16 GB) under Windows 11 with Python 3.10 or later; the library versions of that machine were not recorded. The 600 added pairs and all later covers used the same code and weights on an NVIDIA GeForce RTX 5060 (8 GB) with PyTorch 2.11, Diffusers 0.39.0, Transformers 5.13.0 and PEFT 0.19.1. Embedding, processing, detector training (about 3.4 minutes per epoch) and the rich-model steganalysis ran on that GPU with an AMD Ryzen 5 7500F and Python 3.14; embedding uses only the CPU (NumPy and SciPy, three worker processes).

Use of AI tools

We wrote the original fine-tuning, embedding and evaluation code ourselves. We used ChatGPT (OpenAI) and Claude (Anthropic) to paraphrase and correct the grammar of the manuscript. We checked the derivations, the code and every result ourselves, and we take full responsibility for the content of this article. No AI system is listed as an author.

References

[1] De La Croix, N. J., Ahmad, T. & Han, F. Comprehensive survey on image steganalysis using deep learning. *Array* **22**, 100353 (2024). https://doi.org/10.1016/j.array.2024.100353

[2] Mazumder, J. A. & Hemachandran, K. Study of image steganography using LSB, DFT and DWT. *Int. J. Comput. Technol.* **11**, 2618–2627 (2013). https://doi.org/10.24297/ijct.v11i5.1143

[3] Zakaria, A. A. *et al.* High-capacity image steganography with minimum modified bits based on data mapping and LSB substitution. *Appl. Sci.* **8**, 2199 (2018). https://doi.org/10.3390/app8112199

[4] Bender, W., Gruhl, D., Morimoto, N. & Lu, A. Techniques for data hiding. *IBM Syst. J.* **35**, 313–336 (1996). https://doi.org/10.1147/sj.353.0313

[5] Ahmed, N., Natarajan, T. & Rao, K. R. Discrete cosine transform. *IEEE Trans. Comput.* **C-23**, 90–93 (1974). https://doi.org/10.1109/T-C.1974.223784

[6] Pevný, T., Filler, T. & Bas, P. Using high-dimensional image models to perform highly undetectable steganography. In *Information Hiding*, Lecture Notes in Computer Science **6387**, 161–177 (Springer, 2010). https://doi.org/10.1007/978-3-642-16435-4_13

[7] Holub, V. & Fridrich, J. Designing steganographic distortion using directional filters. In *2012 IEEE International Workshop on Information Forensics and Security (WIFS)* 234–239 (IEEE, 2012). https://doi.org/10.1109/WIFS.2012.6412655

[8] Holub, V., Fridrich, J. & Denemark, T. Universal distortion function for steganography in an arbitrary domain. *EURASIP J. Inf. Secur.* **2014**, 1 (2014). https://doi.org/10.1186/1687-417X-2014-1

[9] Pevný, T., Bas, P. & Fridrich, J. Steganalysis by subtractive pixel adjacency matrix. *IEEE Trans. Inf. Forensics Secur.* **5**, 215–224 (2010). https://doi.org/10.1109/TIFS.2010.2045842

[10] Fridrich, J. & Kodovský, J. Rich models for steganalysis of digital images. *IEEE Trans. Inf. Forensics Secur.* **7**, 868–882 (2012). https://doi.org/10.1109/TIFS.2012.2190402

[11] Kodovský, J., Fridrich, J. & Holub, V. Ensemble classifiers for steganalysis of digital media. *IEEE Trans. Inf. Forensics Secur.* **7**, 432–444 (2012). https://doi.org/10.1109/TIFS.2011.2175919

[12] Boroumand, M., Chen, M. & Fridrich, J. Deep residual network for steganalysis of digital images. *IEEE Trans. Inf. Forensics Secur.* **14**, 1181–1193 (2019). https://doi.org/10.1109/TIFS.2018.2871749

[13] Yu, J., Zhang, X., Xu, Y. & Zhang, J. CRoSS: Diffusion model makes controllable, robust and secure image steganography. In *Advances in Neural Information Processing Systems* **36**, 80730–80743 (2023). https://doi.org/10.52202/075280-3539

[14] Wei, P. *et al.* Generative steganography diffusion. Preprint at https://doi.org/10.48550/arXiv.2305.03472 (2023).

[15] Hu, X. *et al.* Establishing robust generative image steganography via popular Stable Diffusion. *IEEE Trans. Inf. Forensics Secur.* **19**, 8094–8108 (2024). https://doi.org/10.1109/TIFS.2024.3444311

[16] Long, L., Wang, Z. & Zhang, X. Robust image steganography in real social networks using Stable Diffusion. *IEEE Internet Things J.* **13**, 26698–26708 (2026). https://doi.org/10.1109/JIOT.2026.3677876

[17] Mahfuz, A. *et al.* PSyDUCK: Training-free steganography for latent diffusion. Preprint at https://doi.org/10.48550/arXiv.2501.19172 (2025).

[18] Gu, T. *et al.* Enhancing JPEG steganography with GANs via adaptive modification loss and random masking. *IEEE Trans. Circuits Syst. Video Technol.* **36**, 10678–10690 (2026). https://doi.org/10.1109/TCSVT.2026.3674488

[19] Malik, K. R. *et al.* A hybrid steganography framework using DCT and GAN for secure data communication in the big data era. *Sci. Rep.* **15**, 19630 (2025). https://doi.org/10.1038/s41598-025-01054-7

[20] Mendis, T., Kandah, F. & Aakur, S. N. BAFLE-DCT: Bypassing adversarial filters via frequency-selective embedding in the DCT domain. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* 5967–5976 (IEEE, 2026). https://doi.org/10.1109/WACV61042.2026.00577

[21] Kharrazi, M., Sencar, H. T. & Memon, N. Cover selection for steganographic embedding. In *2006 International Conference on Image Processing* 117–120 (IEEE, 2006). https://doi.org/10.1109/ICIP.2006.312386

[22] Black, K., Janner, M., Du, Y., Kostrikov, I. & Levine, S. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations* (2024). Preprint at https://doi.org/10.48550/arXiv.2305.13301

[23] Clark, K., Vicol, P., Swersky, K. & Fleet, D. J. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations* (2024). Preprint at https://doi.org/10.48550/arXiv.2309.17400

[24] Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 10674–10685 (IEEE, 2022). https://doi.org/10.1109/CVPR52688.2022.01042

[25] Hu, E. J. *et al.* LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations* (2022). Preprint at https://doi.org/10.48550/arXiv.2106.09685

[26] Chen, B. & Wornell, G. W. Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. *IEEE Trans. Inf. Theory* **47**, 1423–1443 (2001). https://doi.org/10.1109/18.923725

[27] Radford, A. *et al.* Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, PMLR **139**, 8748–8763 (2021).

[28] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. & Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems* **30** (2017).

[29] Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**, 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861

[30] Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S. & Vedaldi, A. Describing textures in the wild. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 3606–3613 (IEEE, 2014). https://doi.org/10.1109/CVPR.2014.461

[31] Li, B., Wang, M., Huang, J. & Li, X. A new cost function for spatial image steganography. In *2014 IEEE International Conference on Image Processing* 4206–4210 (IEEE, 2014). https://doi.org/10.1109/ICIP.2014.7025854

[32] Sedighi, V., Cogranne, R. & Fridrich, J. Content-adaptive steganography by minimizing statistical detectability. *IEEE Trans. Inf. Forensics Secur.* **11**, 221–234 (2016). https://doi.org/10.1109/TIFS.2015.2486744

[33] Ker, A. D. & Böhme, R. Revisiting weighted stego-image steganalysis. In *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, Proc. SPIE **6819**, 681905 (2008). https://doi.org/10.1117/12.766820

[34] Qian, Y., Dong, J., Wang, W. & Tan, T. Deep learning for steganalysis via convolutional neural networks. In *Media Watermarking, Security, and Forensics 2015*, Proc. SPIE **9409**, 94090J (2015). https://doi.org/10.1117/12.2083479

[35] Cogranne, R., Giboulot, Q. & Bas, P. Steganography by minimizing statistical detectability: the cases of JPEG and color images. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security* 161–167 (ACM, 2020). https://doi.org/10.1145/3369412.3395075

[36] Holub, V. & Fridrich, J. Low-complexity features for JPEG steganalysis using undecimated DCT. *IEEE Trans. Inf. Forensics Secur.* **10**, 219–228 (2015). https://doi.org/10.1109/TIFS.2014.2364918

[37] Ker, A. D. *et al.* Moving steganography and steganalysis from the laboratory into the real world. In *Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security* 45–58 (ACM, 2013). https://doi.org/10.1145/2482513.2482965

[38] Kodovský, J., Sedighi, V. & Fridrich, J. Study of cover source mismatch in steganalysis and ways to mitigate its impact. In *Media Watermarking, Security, and Forensics 2014*, Proc. SPIE **9028**, 90280J (2014). https://doi.org/10.1117/12.2039693

[39] Giboulot, Q., Cogranne, R., Borghys, D. & Bas, P. Effects and solutions of cover-source mismatch in image steganalysis. *Signal Process. Image Commun.* **86**, 115888 (2020). https://doi.org/10.1016/j.image.2020.115888

[40] Cogranne, R., Giboulot, Q. & Bas, P. ALASKA#2: Challenging academic research on steganalysis with realistic images. In *2020 IEEE International Workshop on Information Forensics and Security (WIFS)* 1–5 (IEEE, 2020). https://doi.org/10.1109/WIFS49906.2020.9360896

[41] Mongaras, G. Imagenet21K_Recaption [dataset]. Hugging Face https://huggingface.co/datasets/gmongaras/Imagenet21K_Recaption (accessed 1 October 2026).

[42] Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems* **33**, 6840–6851 (2020).

[43] Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations* (2019). Preprint at https://doi.org/10.48550/arXiv.1711.05101

[44] Song, J., Meng, C. & Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations* (2021). Preprint at https://doi.org/10.48550/arXiv.2010.02502

[45] Filler, T., Judas, J. & Fridrich, J. Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Trans. Inf. Forensics Secur.* **6**, 920–935 (2011). https://doi.org/10.1109/TIFS.2011.2134094

[46] Qian, Y., Dong, J., Wang, W. & Tan, T. Learning and transferring representations for image steganalysis using convolutional neural network. In *2016 IEEE International Conference on Image Processing* 2752–2756 (IEEE, 2016). https://doi.org/10.1109/ICIP.2016.7532860

Author contributions

M.E.H.M.: Writing (original draft), Software, Methodology, Investigation, Formal analysis, Conceptualization. Designed and implemented the DCT-based dataset filtering pipeline and the adaptive Y-channel DCT steganography framework, including distortion-aware payload placement and parity embedding via quantisation-based slot ordering. I.R.: Software, Methodology, Investigation, Conceptualization. Originated the core hypothesis that biasing generated images toward stronger DCT coefficients improves embedding capacity; proposed Stable Diffusion v1.5 as

the backbone; implemented the preprocessing pipeline, the LoRA configuration on U-Net attention projections, the DCT reward loss, the frozen anchor U-Net regularisation, and the full training loop. M.H.: Software, Methodology, Investigation. Designed the convolutional steganalyser architecture with adjusted SRM filter kernels and trained its earlier versions; implemented competing steganalysers for benchmarking; implemented the canonical Huffman compression module. S.T.C.: Writing (original draft), Visualization, Investigation, Formal analysis. Compared DEDS against existing methods; tested pipeline outputs and evaluated stego image results across all cover images; produced the diagrams of the original manuscript; prepared the article in LaTeX. M.F.J.: Supervision, Methodology. Provided guidance on the experimental design and helped refine the methodology throughout the project. M.I.H.: Validation, Supervision, Resources, Project administration, Methodology, Conceptualization. All authors reviewed the manuscript.

Data availability

The fine-tuning and dataset filtering code is at `https://github.com/Imraj-Rabbani/P3-Image-Steganography-Pipeline`. The embedder, generation and evaluation scripts, prompt bank, LoRA adapter weights, 1,000-pair evaluation corpus, cover statistics, training records and per-image results are at `https://github.com/Ezazulmahi/DEDS`, with the stego images attached to release v2.0. The steganalysis code, settings, logs and per-image scores are in its `steganalysis` folder, with training images, trained detectors and features attached to release v3.0. The training images come from the public `gmongaras/Imagenet21K_Recaption` dataset and can be regenerated with the released filtering script. The Lorem Picsum photographs used for the spectral comparison are listed by identifier in the repository.

Code availability

All code used for the results, including fine-tuning, embedding, evaluation, steganalysis and extra checks, is in the two repositories above (releases v2.0 and v3.0 of the second), with package versions in `requirements.txt`, under the MIT licence without restrictions.

Funding

This research was supported by BRAC University, Dhaka, Bangladesh, through institutional research support provided to the Department of Computer Science and Engineering.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information (Figs. S1 to S3, Tables S1 to S18, Notes 1 to 4) accompanies this paper. **Correspondence** should be addressed to M.I.H.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Suppliment.pdf