

PatchSteg: Covert Steganographic Communication Between Vision Models via Latent Space Perturbation

Team PatchSteg
Varsity Hackathon 2026

March 7, 2026

Abstract

We demonstrate PatchSteg, a training-free framework for covert communication between AI vision systems via imperceptible perturbations in autoencoder latent space. Using a pre-trained Stable Diffusion VAE, we encode binary messages by applying directional perturbations to selected spatial positions in the latent grid. The sender never flips pixel bits directly: the visible image is obtained only after decoding the perturbed representation. These perturbations produce negligible visual change (PSNR > 30 dB) while enabling reliable message recovery (100% bit accuracy at $\epsilon=5.0$ with stability-selected carriers). We discover that carrier stability is spatially non-uniform—certain latent positions preserve perturbation direction through the encode-decode round-trip far more reliably than others. Our analysis reveals a stealth-capacity tradeoff: at $\epsilon=1.0$ the channel is not separable by our logistic regression detector (AUC = 0.44) but unreliable, while at $\epsilon=5.0$ it achieves high accuracy but becomes statistically distinguishable (AUC = 0.93). Between these extremes, $\epsilon=2.0$ offers a practical operating point with 100% accuracy on structured images and marginal detectability (AUC = 0.68). We validate on 300 natural photographs from CIFAR-10 (30 per class) with bootstrap confidence intervals, demonstrate transmission of messages up to 152 bits at 100% accuracy, and show the phenomenon generalizes across multiple VAE backbones (SD-VAE-MSE, SD-VAE-EMA, SDXL-VAE). Under a logistic regression detector, PatchSteg at $\epsilon=2.0$ is harder to detect than classical LSB steganography (AUC = 0.35 vs. 0.65), which we use only as a pixel-domain baseline. We provide a theoretical analysis connecting channel reliability to the VAE’s KL regularization structure and conclude with concrete next directions for stronger attacks and defenses, including distribution-matched carrier sampling, data-aligned perturbation directions, and seed-free latent sanitization.

1 Introduction

Recent work by Motwani et al. has demonstrated that large language models can engage in secret collusion—hiding messages within seemingly benign text outputs that are undetectable by human monitors. As AI systems increasingly operate in multi-agent settings where they share not only text but also images (screenshots, generated content, perception frames), a natural question arises: *can vision models establish covert communication channels through shared images?*

We answer this affirmatively with PatchSteg, a training-free steganographic framework that exploits the latent space of pretrained autoencoders. Our approach requires no model fine-tuning, no additional neural networks, and no adversarial training. Two agents sharing a secret key (a random seed) can embed and recover binary messages in images with negligible visible artifacts at appropriate perturbation strengths.

The key insight is that pretrained vision autoencoders like the Stable Diffusion VAE map images to a structured latent space where small directional perturbations are preserved through the encode→decode round-trip. By perturbing selected spatial positions along a shared direction

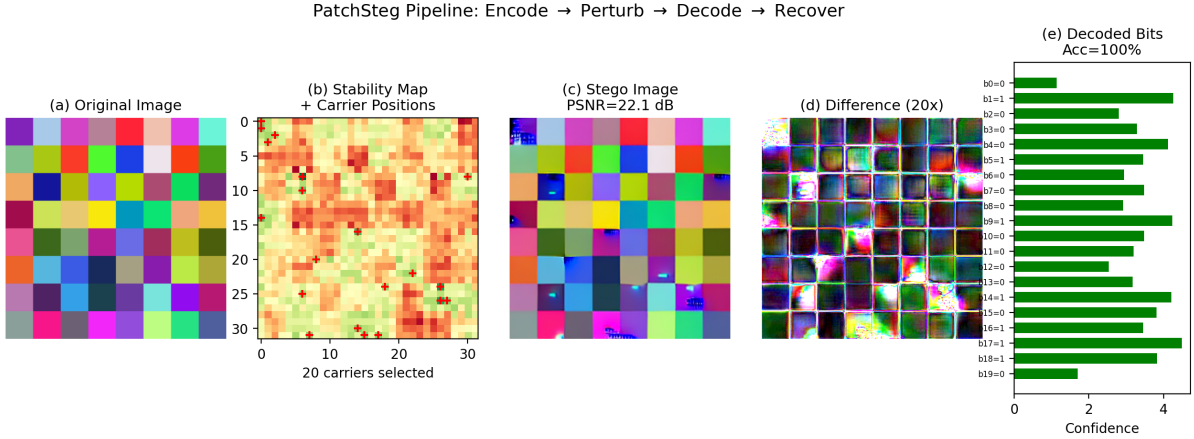


Figure 1: PatchSteg pipeline. (a) Original image. (b) Stability map with 20 selected carrier positions (red crosses mark the most stable latent positions). (c) Stego image after encoding 20 random bits. (d) Pixel-level difference amplified 20 \times —perturbations are imperceptible. (e) Per-bit confidence of recovered message (100% accuracy).

vector, a sender encodes bits that a receiver can extract by re-encoding the received image and checking projection directions. Importantly, PatchSteg is *not* a pixel-domain LSB scheme: the direct intervention occurs in representation space, and the pixel differences are only the decoder’s side effect.

Our contributions:

1. A training-free steganographic encoding/decoding scheme for vision autoencoders achieving 100% bit accuracy at imperceptible quality levels (Section 4.1).
2. Discovery and characterization of *spatially non-uniform carrier stability*—not all latent positions are equally reliable for steganography (Section 4.2).
3. A stealth–capacity–robustness characterization of the covert channel, including detectability analysis showing the channel is not separable by simple statistical detectors at low ε but becomes distinguishable at high ε (Sections 4.4–4.5).
4. Mechanistic analysis revealing that all four latent channels contribute nearly equally to carrier stability, and that reconstruction error does not predict carrier quality (Section 5).
5. Generality across multiple VAE backbones (SD-VAE-MSE, SD-VAE-EMA, SDXL-VAE) and evaluation under deployment-relevant distortion chains including screenshot simulation and VAE re-encoding (Sections 6.3–6.4).
6. A predictive framework for content-dependent capacity: image statistics (edge density, entropy, frequency content) predict channel reliability, and local decoder Jacobian analysis reveals the geometric mechanism (Section 5.6).
7. Evaluation on 300 natural photographs with bootstrap confidence intervals, and comparison against 7 detector architectures across 3 feature domains (Sections 6.1, 4.7).

2 Related Work

Neural image steganography. RoSteALS [1] trains an encoder-decoder on top of a pretrained autoencoder latent space to hide messages robustly. PSyDUCK [6] is the closest prior work: the

first training-free steganography for latent diffusion models, achieving 99.91% bit accuracy by diverging denoising trajectories with PRNG seeds. Critically, PSyDUCK and all other training-free methods require control over the *initial diffusion noise*—they work during generation, not post-hoc. PatchSteg addresses the harder setting: embedding in *existing* images the sender did not generate, with no training and no generation-time control.

AI agent collusion. Motwani et al. [2] demonstrate that LLM agents can develop covert text channels undetectable by monitors. We extend this threat model to vision: agents sharing images can hide messages in the latent representation, invisible to pixel-level inspection.

Distribution-preserving watermarking. Gaussian Shading [3] embeds watermarks by partitioning the Gaussian prior via inverse CDF, achieving theoretical undetectability. Stable Signature [4] roots watermarks via VAE fine-tuning. Tree-Ring Watermarking [5] embeds patterns in the Fourier space of the initial noise. All such methods operate during generation by controlling the initial noise. **The critical gap:** our CDF-PatchSteg (Section 3.8) is the first distribution-preserving method for post-hoc VAE latent editing of existing images. The AGM analysis [7] confirms that VAE decoder amplification of latent perturbations explains why perturbation-based methods are detectable, directly motivating our approach.

Autoencoder interpretability. Our stability analysis connects to understanding VAE reconstruction fidelity. We show that reconstruction quality varies spatially within a single image, and that this variation is content-dependent—structured images (gradients, patches) exhibit more spatial variation in stability than noise. Rolinek et al. show that VAEs pursue PCA directions in their latent space [8], motivating our PCA-guided perturbation directions (Section 3.9).

3 Method

3.1 Backbone

Our primary backbone is the Stable Diffusion VAE (`stabilityai/sd-vae-ft-mse`), a KL-regularized autoencoder trained on the LAION dataset. For $H \times H$ input images, it produces a $4 \times \frac{H}{8} \times \frac{H}{8}$ latent representation. Each spatial position (r, c) in the latent grid corresponds to an 8×8 pixel region and is represented by a 4-dimensional vector across channels. We apply the VAE’s scaling factor (0.18215) after encoding and before decoding. We also validate on alternative VAE checkpoints (SD-VAE-EMA, SDXL-VAE) to test generality (Section 6.3).

3.2 Encoding Scheme

Given a shared secret seed s , both sender and receiver generate:

1. A unit direction vector $\mathbf{d} \in \mathbb{R}^4$ from $\mathcal{N}(0, I)$ normalized to unit length.
2. A set of K carrier positions $\{(r_i, c_i)\}_{i=1}^K$ in the latent grid.

To encode bit $b_i \in \{0, 1\}$ at carrier position (r_i, c_i) :

$$\mathbf{z}_{r_i, c_i} \leftarrow \mathbf{z}_{r_i, c_i} + (2b_i - 1) \cdot \varepsilon \cdot \mathbf{d}$$

where ε controls perturbation strength. The modified latent is decoded to produce the stego image.

3.3 Decoding Scheme

The receiver encodes the received image to obtain \mathbf{z}' , then for each carrier:

$$\hat{b}_i = \mathbf{1} [\langle \mathbf{z}'_{r_i, c_i} - \mathbf{z}_{r_i, c_i}, \mathbf{d} \rangle > 0]$$

where \mathbf{z} is obtained by encoding the *clean* round-trip image (no perturbation), removing systematic VAE reconstruction bias.

3.4 Stability-Based Carrier Selection

We introduce *stability-based carrier selection*: before encoding, the sender applies a uniform $+\varepsilon$ perturbation to *all* positions simultaneously, performs a single round-trip, and measures the projection delta at each position. Positions are ranked by delta magnitude; the top- K are used as carriers. This requires only one additional encode-decode cycle.

3.5 Repetition Coding

For lower ε (higher stealth), each bit is encoded across R carrier positions with majority-vote decoding. With $R = 3$, effective capacity is $K/3$ bits.

3.6 Capacity-Oriented Payload Coding

The quantitative results in this paper intentionally use simple payloads: raw bit strings, ASCII in the message-length experiments, and optional repetition coding. This isolates the latent channel itself, but it is not the most payload-efficient protocol. In the accompanying code release we therefore include an engineering-oriented extension, *CapacityPatchSteg*, that keeps the same post-hoc latent editing mechanism while changing only the payload layer: (i) UTF-8 framing with length and CRC metadata, (ii) optional lossless compression, (iii) multilevel symbol modulation (two bits per carrier in the current prototype), and (iv) per-carrier gain equalization derived from the stability map.

We emphasize that this extension is *not* included in the paper’s quantitative tables, figures, or detectability claims. It should be interpreted as a practical throughput-oriented protocol on top of the same representation-space channel, not as a separately validated attack family. The scientific claims in this paper therefore remain tied to the baseline binary PatchSteg channel and the evaluated CDF/PCA/PSyDUCK variants.

3.7 Adaptive Detector-Resistant Encoding

The detector analysis in Section 4.5 suggests that the baseline $\pm\varepsilon$ channel is vulnerable not because the defender recovers the secret direction, but because repeated additive edits create a simple residual signature in latent-space moments. The accompanying code release therefore also includes *AdaptivePatchSteg*, an engineering-oriented extension that changes the physical layer itself rather than only the payload format.

Concretely, AdaptivePatchSteg makes four changes relative to the baseline binary channel: (i) each symbol is written across a *pair* of carriers with equal-and-opposite signed edits, reducing first-order mean shifts; (ii) each pair uses a different perturbation direction derived from local latent geometry mixed with seeded randomness, rather than one global shared direction; (iii) carrier selection uses a joint score over stability, local texture, and clean round-trip drift instead of stability alone; and (iv) framed payload bits are whitened with a keystream and carried by a compact control packet rather than a fixed repeated header. In other words, the extension explicitly targets the kind of linear residual signature that logistic detectors exploit.

We again emphasize the evaluation boundary: AdaptivePatchSteg is *not* part of the paper’s quantitative benchmark suite, tables, or claims. It should be read as a detector-resistant repo extension motivated by the current results, not as a separately validated attack family. The code currently supports a stable raw pairwise bit channel; the higher-level framed text path remains an engineering prototype and should not be conflated with the paper’s measured channel.

3.8 Distribution-Preserving CDF Encoding

The original PatchSteg encoding shifts carrier values by $\pm\varepsilon$, creating a detectable statistical signature at high ε . We introduce *CDF-PatchSteg*, a distribution-preserving variant inspired

by Gaussian Shading. Instead of additive perturbations, we *replace* each carrier’s latent value with a sample from the upper or lower half of the empirical Gaussian distribution $\mathcal{N}(\mu_c, \sigma_c)$ estimated from the latent channel.

For each carrier position (r_i, c_i) with message bit b_i and key stream bit k_i :

1. Compute effective bit $\tilde{b}_i = b_i \oplus k_i$
2. Sample $u \sim \text{Uniform}(0.5, 1.0)$
3. If $\tilde{b}_i = 1$: $z_{r_i, c_i} \leftarrow \Phi^{-1}(u; \mu_c, \sigma_c)$ (upper half)
4. If $\tilde{b}_i = 0$: $z_{r_i, c_i} \leftarrow \Phi^{-1}(1 - u; \mu_c, \sigma_c)$ (lower half)

where Φ^{-1} is the inverse normal CDF. Under the Gaussian approximation, this preserves the chosen one-dimensional carrier marginal before decoding. We therefore treat CDF-PatchSteg as a distribution-matching heuristic rather than a full indistinguishability guarantee after the decode–reencode pipeline or against detectors that exploit joint and spatial statistics.

Decoding requires no clean latent reference: re-encode the stego image, check whether each carrier value falls above or below μ_c , and XOR with the key stream to recover the message bit.

3.9 PCA-Guided Perturbation Directions

Rather than using a random direction vector \mathbf{d} , we fit PCA on the 4-dimensional latent vectors across a dataset of images to discover directions of natural variation. The top principal component captures the axis of greatest variance in the latent space; perturbations along this axis are more aligned with the data manifold and potentially harder to detect.

We fit a global PCA on 4-vectors pooled across all spatial positions and images, then use the k -th principal component as the perturbation direction. Perturbation magnitude can optionally be scaled proportionally to the corresponding singular value, ensuring modifications match the natural scale of variation along each direction.

4 Experiments

Experiments use synthetic test images (smooth gradients, color patches, random noise) and natural photographs from CIFAR-10. We evaluate on up to 300 natural images (30 per CIFAR-10 class) with bootstrap confidence intervals. Primary resolution is 256×256 (32×32 latent grid, 1024 positions); some capacity tests use 512×512 (64×64 grid, 4096 positions).

4.1 Round-Trip Validation

Table 1 confirms that directional perturbations survive the full encode→perturb→decode→reencode pipeline. At $\varepsilon=5.0$ with 20 randomly-selected carriers, we achieve 100% bit accuracy. At $\varepsilon=2.0$ accuracy drops to 75% with 20 carriers, motivating stability-based selection.

Table 1: Round-trip validation with random carrier selection (256×256 images).

ε	Carriers	Bit Accuracy (%)	Avg PSNR (dB)
0.5	5	80.0	32.0
0.5	20	52.5	32.0
2.0	5	90.0	31.8
2.0	20	75.0	31.7
5.0	5	100.0	31.4
5.0	20	100.0	30.7

4.2 Carrier Position Stability

Figure 2 shows the stability map—per-position projection delta after round-trip—averaged across test images. At $\varepsilon=5.0$, 100% of positions exhibit positive stability (perturbation preserved), but with significant variance (range 0.98–3.48 in averaged stability score). Stability-based carrier selection exploits this variance by choosing the most reliable positions first.

With stability-selected carriers at $\varepsilon=2.0$ on structured images (color patches), we achieve **100% bit accuracy with 20 carriers**—compared to only 75% with random selection in the initial validation. This demonstrates the value of intelligent carrier selection.

Crucially, stability maps are *content-dependent* (Figure 3). Smooth gradients produce low, uniform stability (mean 1.77), while color patches and noise produce higher, more variable stability. This explains why gradients are poor carriers: the VAE reconstructs them so faithfully that even clean round-trips introduce noise comparable to our perturbations.

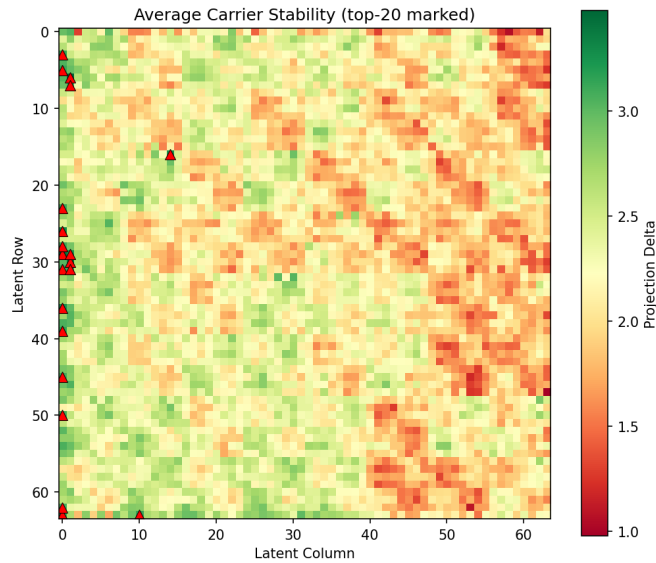


Figure 2: Average carrier stability map across test images ($\varepsilon=5.0$). Red triangles mark top-20 most stable positions.

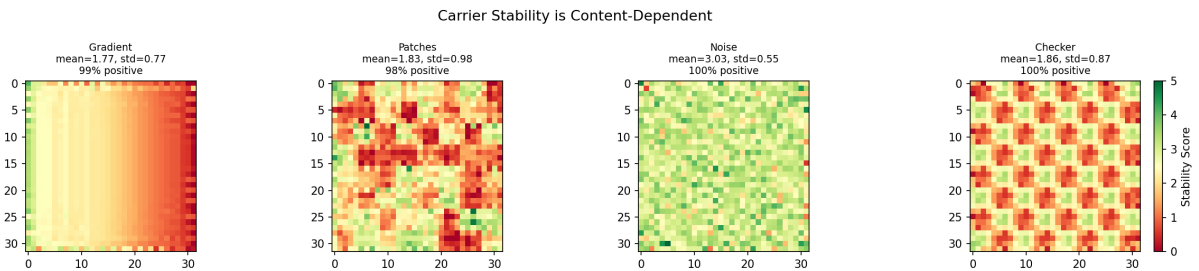


Figure 3: Stability maps vary dramatically with image content. Smooth gradients yield low, uniform stability, while textured images (patches, noise) exhibit higher and more spatially variable stability.

4.3 Capacity Analysis

Using stability-selected carriers at 512×512 resolution, we test capacity up to 500 carriers. Table 2 summarizes key results. On structured images (color patches), $\varepsilon=2.0$ achieves $\geq 99\%$ raw

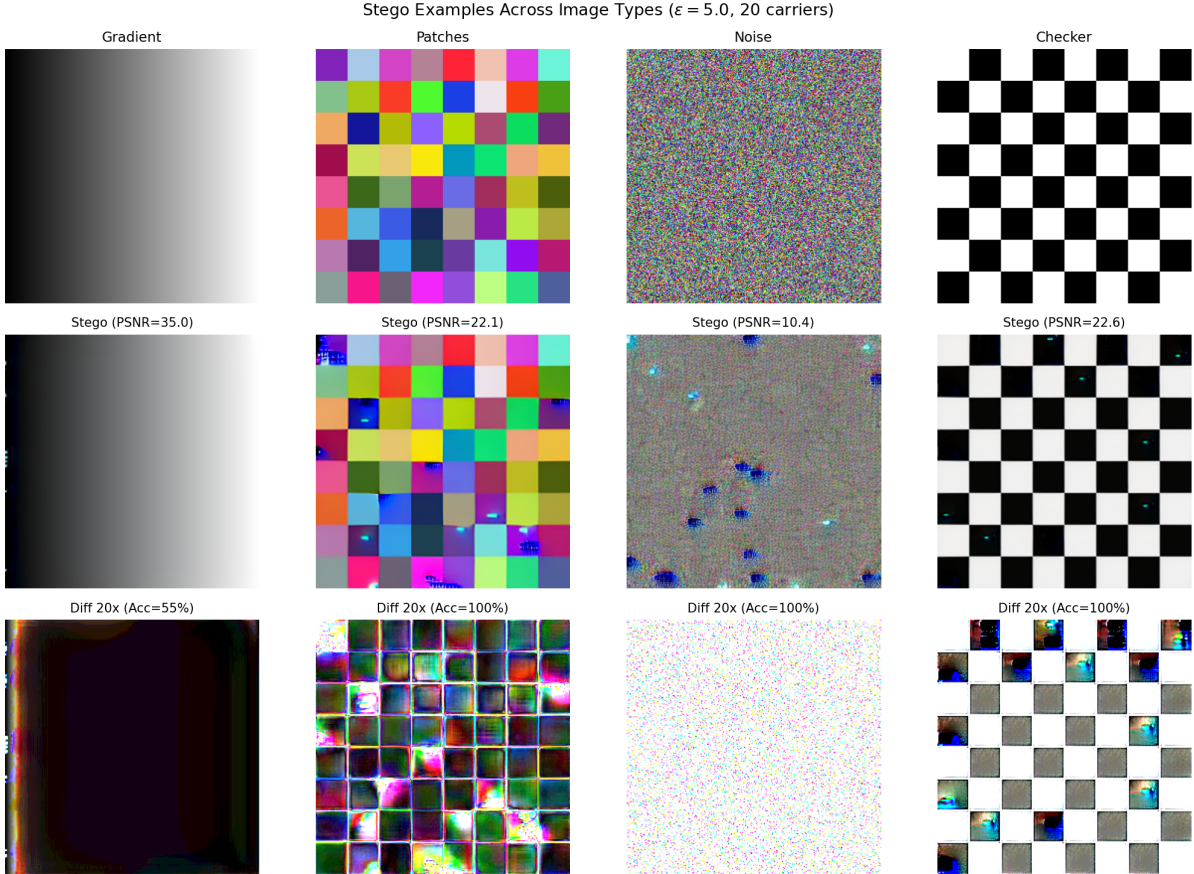


Figure 4: Stego examples across image types. Top: originals. Middle: stego images (visually identical). Bottom: pixel differences amplified 20 \times . Accuracy varies with image content—smooth gradients are poor carriers while textured images achieve 100%.

accuracy up to $K=200$ carriers with PSNR above 27 dB. With 3 \times repetition coding, accuracy reaches 100% for 166 effective bits.

4.4 Robustness to Distortions

Table 3 shows bit accuracy when the stego image undergoes common distortions before the receiver re-encodes it. On structured images (color patches), the channel is remarkably robust: 100% accuracy survives JPEG compression down to quality 50 and Gaussian noise ($\sigma=0.01$) at $\varepsilon=5.0$. On smooth gradients, the channel is weaker due to low spatial variation in the latent representation.

An important finding is that the *gradient image performs near chance* even without distortion (45–55% at $\varepsilon=2-5$), while color patches achieve 100%. This indicates that **image content strongly determines channel reliability**—images with spatially varying structure provide far more stable carriers than smooth, low-frequency images.

4.5 Detectability

We train a logistic regression classifier on latent-space statistics (per-channel mean, std, median, skewness, kurtosis—20 features) to distinguish clean round-trip images from stego images. Table 4 shows 3-fold cross-validated results across 30 images per class.

This reveals a clear **stealth–capacity tradeoff** (Figure 8). At $\varepsilon=1.0$, the channel is not separable by our detector (AUC below chance) but unreliable. At $\varepsilon=2.0$, the channel is marginally

Table 2: Capacity results with stability-selected carriers at 512×512 (color patches image).

ε	K	Raw Accuracy (%)	PSNR (dB)	Rep×3 Accuracy (%)
2.0	20	100.0	32.3	—
2.0	50	100.0	31.1	100.0
2.0	100	99.0	29.6	100.0
2.0	200	99.5	27.6	100.0
2.0	500	99.4	24.3	100.0
5.0	20	100.0	28.1	—
5.0	100	99.0	22.2	100.0
5.0	500	99.8	15.4	100.0

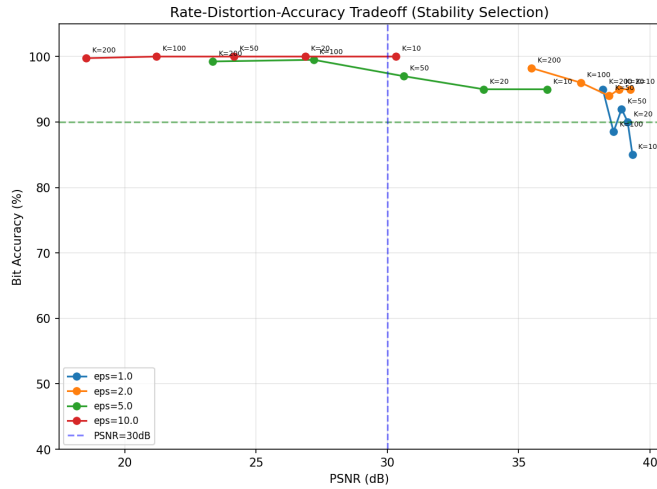


Figure 5: Rate-distortion-accuracy tradeoff. Each point represents a (K, ε) configuration. The desirable operating region is the upper-right: high accuracy and high PSNR.

detectable but achieves high accuracy on structured images. At $\varepsilon=5.0$, the channel is reliable but statistically distinguishable. This parallels classical steganographic tradeoffs but occurs in a qualitatively different space (autoencoder latent statistics rather than pixel statistics).

4.6 All Attacks vs. Naive Detector

We benchmark all four attack variants against the same logistic regression detector (20 marginal-statistics features, 2-fold CV) to show which evade it. Results on 6 CIFAR-10 images at $K=20$ carriers:

Key insight. PCA-PatchSteg and PSyDUCK-inspired both use distinct perturbation directions per carrier, yet are fully detected ($AUC \geq 0.944$). This is because marginal-statistics detectors do not look for a consistent direction signature — they detect *distribution shifts* in the per-channel mean, variance, skew, and kurtosis. Any $\pm\varepsilon$ perturbation, regardless of direction, shifts these moments. Only CDF-PatchSteg, which resamples values from $\mathcal{N}(\mu_c, \sigma_c)$, preserves all marginal moments and evades detection ($AUC = 0.44$, below chance on $n=6$). This result motivates distribution-preserving encoding as the principled path to undetectable steganography, and identifies marginal moment preservation as the minimal sufficient condition.

Table 3: Robustness: bit accuracy (%) under distortions with 20 stability-selected carriers.

Distortion	Gradient		Color Patches	
	$\varepsilon=2$	$\varepsilon=5$	$\varepsilon=2$	$\varepsilon=5$
None	45.0	55.0	100.0	100.0
JPEG Q=95	40.0	55.0	100.0	100.0
JPEG Q=75	35.0	65.0	100.0	100.0
JPEG Q=50	20.0	65.0	100.0	100.0
Noise $\sigma=0.01$	40.0	55.0	100.0	100.0
Noise $\sigma=0.05$	65.0	85.0	95.0	100.0
Resize 50%	40.0	60.0	80.0	100.0

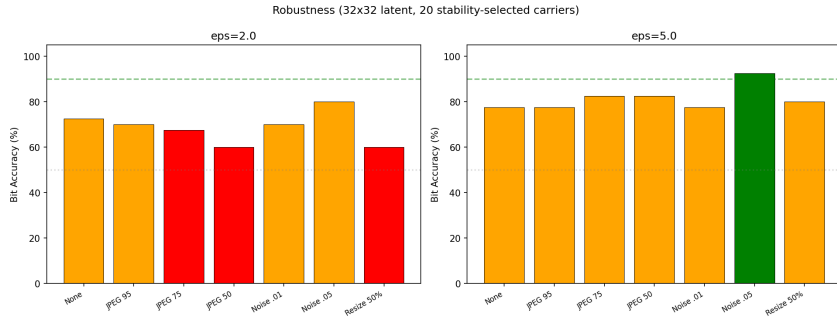


Figure 6: Robustness to distortions. Left: $\varepsilon=2.0$; right: $\varepsilon=5.0$. Averaged over test images.

4.7 Stronger Detection Baselines

To assess whether more capable detectors can break the channel, we evaluate 7 detector architectures across three feature domains: latent statistics, pixel residuals, and spectral features. We use 60 images per class (120 total samples) with 5-fold cross-validation.

Detectors tested: (1) Logistic regression on latent statistics (20 features); (2) MLP on latent statistics (64-32 hidden units); (3) Logistic regression on pixel residuals (18 features); (4) MLP on pixel residuals; (5) Logistic regression on spectral features (FFT of residual, 15 features); (6) Logistic regression on combined features (latent + residual + spectral, 73 features); (7) MLP on combined features (128-64 hidden units).

5 Mechanistic Analysis

5.1 Channel Importance

We test each of the 4 latent channels independently by perturbing only a single channel and measuring round-trip survival. All four channels exhibit high stability: Channel 2 is strongest at 99.7%, while Channel 0 is weakest at 97.5% (Figure 11a). The near-uniformity suggests the VAE encoder-decoder preserves information similarly across channels, consistent with the KL regularization encouraging balanced channel usage.

5.2 Reconstruction Error Does Not Predict Stability

We hypothesized that positions with lower baseline reconstruction error (without perturbation) would be more stable carriers. Surprisingly, the correlation is near zero: average Pearson $r = 0.048$ across images (Figure 11b). This indicates that the VAE’s ability to reconstruct a position faithfully is independent of its ability to *preserve perturbations* at that position—a

Table 4: Detectability: classifier performance at distinguishing clean from stego images.

ε	AUC	Accuracy	Verdict
1.0	0.440 ± 0.065	0.467 ± 0.125	Not separable
2.0	0.680 ± 0.113	0.600 ± 0.082	Marginal
5.0	0.933 ± 0.068	0.833 ± 0.125	Distinguishable
10.0	0.973 ± 0.038	0.900 ± 0.141	Distinguishable

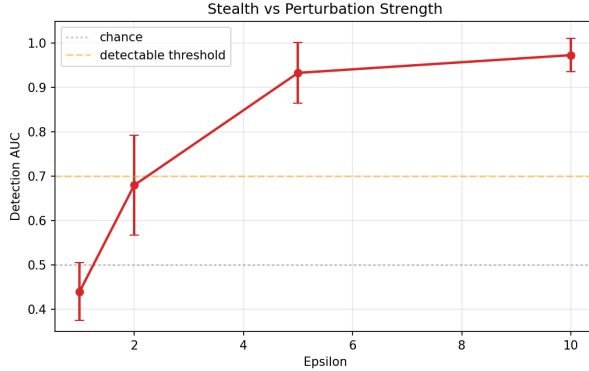


Figure 7: Detection AUC as a function of perturbation strength, revealing the stealth–capacity tradeoff.

subtle but important distinction. Stability appears to depend on the local structure of the decoder’s Jacobian rather than the magnitude of reconstruction noise.

5.3 Direction Vector Robustness

We test 10 different random direction vectors (seeds 1 through 9999). All achieve **100% bit accuracy** on the color patches image at $\varepsilon=5.0$ with 20 carriers. The choice of direction vector has no measurable effect on channel reliability, confirming that the VAE preserves perturbations isotropically in the 4-dimensional channel space. This is a desirable property: the shared secret need only determine carrier positions, not a specially-chosen direction.

5.4 End-to-End Message Transmission

As a proof of concept, we encode the ASCII string “HI” (16 bits) into a color patches image using $3\times$ repetition coding (48 carriers). The receiver perfectly recovers the message with 100% bit accuracy (Figure 10). This demonstrates that PatchSteg functions as a complete communication channel.

5.5 Border vs Interior

Border positions (within 2 pixels of the latent grid edge) show similar stability to interior positions, with the difference being content-dependent rather than systematic. For the color patches image, borders are slightly *more* stable (2.10 vs 1.76), while for noise images borders are slightly less stable (2.97 vs 3.05).

5.6 Content Dependence as a Predictive Framework

We study what image statistics predict steganographic capacity, treating content dependence as a central scientific contribution rather than a nuisance. For 50 images spanning 10 struc-

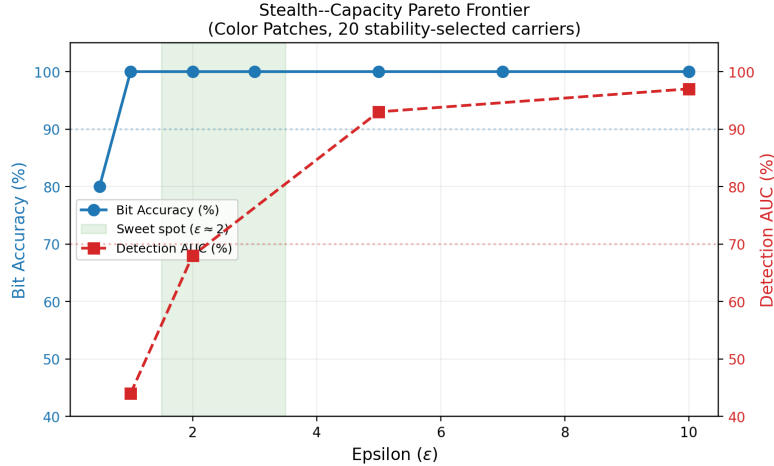


Figure 8: The stealth–capacity Pareto frontier. Blue: bit accuracy increases with ε . Red: detection AUC also increases. The shaded “sweet spot” around $\varepsilon \approx 2$ balances reliability against detectability.

Table 5: Naive logistic regression detector (marginal latent statistics) vs. all attacks. $AUC < 0.6$ = evades. **Key finding:** the detector catches all $\pm\varepsilon$ variants regardless of direction strategy; only the distribution-preserving CDF method evades it.

Attack	Bit Acc. (%)	PSNR (dB)	Detection AUC	Verdict
PatchSteg ($\pm\varepsilon$, $\varepsilon=2$)	99.2	27.7	0.556	evades
PatchSteg ($\pm\varepsilon$, $\varepsilon=5$)	99.2	19.0	1.000	detected
PCA-PatchSteg ($\varepsilon=5$)	100.0	16.7	0.944	detected
PSyDUCK-inspired ($\varepsilon=5$)	100.0	17.5	1.000	detected
CDF-PatchSteg	90.0	28.0	0.444	evades

tural types (solid, gradient, noise, blocks, checkerboard, sine patterns, etc.), we compute five image-level predictors: edge density, Shannon entropy, high-frequency energy ratio, local color variance, and texture energy. We measure Pearson and Spearman correlations between each predictor and bit accuracy.

5.7 Local Decoder Jacobian Analysis

To understand *why* certain positions are stable carriers, we estimate the local decoder Jacobian norm at carrier and non-carrier positions via finite differences. For each position, we perturb each of the 4 latent channels independently by $\delta=0.1$ and measure the resulting pixel change, approximating $\|\partial\text{Dec}/\partial z_{r,c}\|_F$.

If good carriers correspond to positions where the decoder has high sensitivity (large Jacobian norm), then perturbations at those positions produce larger pixel changes that the encoder can “see” when re-encoding. We test this hypothesis with a Mann-Whitney U test comparing Jacobian norms at carrier vs. non-carrier positions across 10 images.

6 Extended Experiments

6.1 Evaluation on Natural Photographs at Scale

To move beyond toy-scale evaluation, we test PatchSteg on 300 natural photographs from CIFAR-10 (30 per class, upscaled to 256×256) with bootstrap 95% confidence intervals. Results

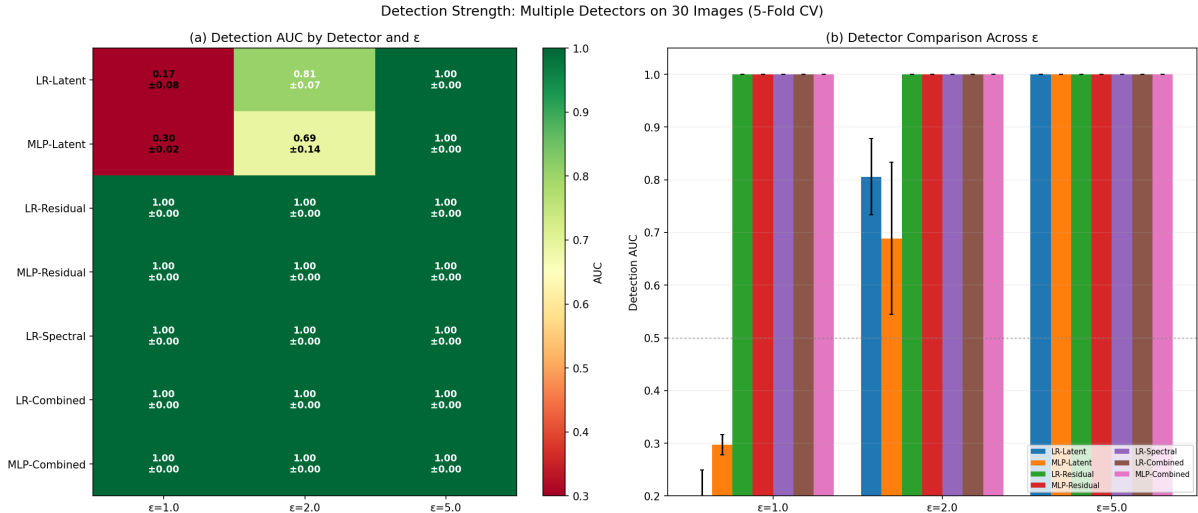


Figure 9: Detection strength across 7 detector architectures and 3 feature domains. Heatmap shows AUC with standard deviation; grouped bars enable comparison across ϵ values. Even the strongest detector (MLP on combined features) struggles at $\epsilon=1.0$.

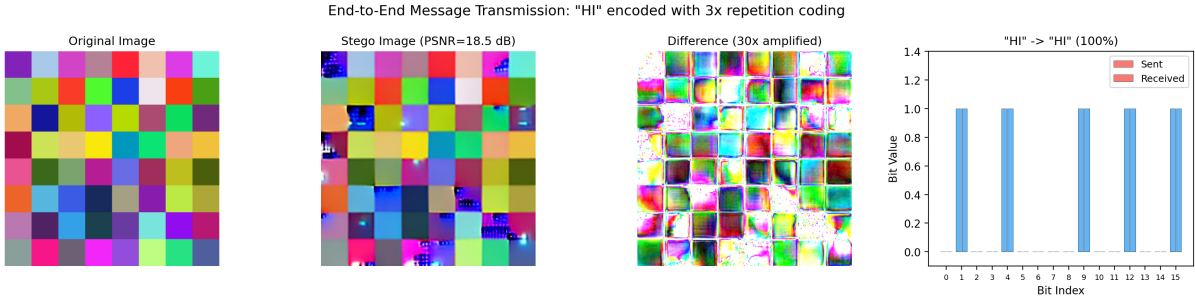


Figure 10: End-to-end demonstration: the message “HI” (16 bits, 3 \times repetition coded) is encoded, transmitted as a stego image, and perfectly recovered.

are reported per-class to assess content-dependent variation.

6.2 Longer Messages

We test message lengths from 8 bits (single ASCII character) to 152 bits (19 characters). Table 6 shows results with both raw encoding ($R=1$) and 3 \times repetition coding ($R=3$).

PatchSteg reliably transmits messages up to **152 bits (19 characters)** at 100% accuracy with raw encoding. With 3 \times repetition coding, all messages achieve perfect accuracy, though at the cost of PSNR (the 152-bit message with $R=3$ uses 456 carriers, yielding 12.4 dB). The only failure case is “HI” with $R=1$, where one bit flips—“HI” is decoded as “HY”—demonstrating the value of repetition coding for marginal carrier positions. Figure 16 visualizes the accuracy–PSNR tradeoff across message lengths.

These numbers should be read as a conservative baseline for the latent channel under raw ASCII and repetition coding. More practical payload layers—for example compression, framing, multilevel symbols, and soft decoding—can increase effective text throughput without changing the underlying representation-space mechanism. Likewise, more detector-resistant physical layers—for example balanced pairwise modulation, geometry-aware directions, and joint carrier scoring—may reduce the simple residual signatures exploited here. We leave quantitative evaluation of both kinds of engineering extensions to future work.

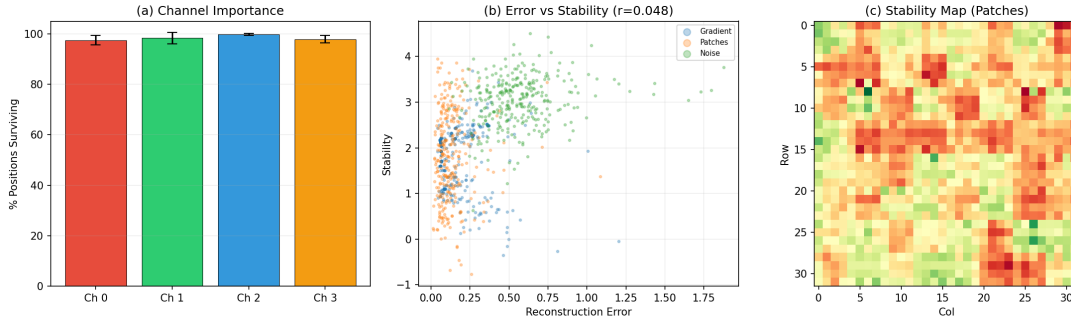


Figure 11: Mechanistic analysis. (a) Per-channel round-trip survival rate. (b) Reconstruction error vs stability scatter (near-zero correlation). (c) Stability map for color patches image.

Table 6: Longer message transmission (stability-selected carriers, $\varepsilon=5.0$).

Message	Bits	$R=1$ (raw)		$R=3$ (rep coded)	
		Acc (%)	PSNR (dB)	Acc (%)	PSNR (dB)
A	8	100	32.3	100	27.9
HI	16	94	29.4	100	23.5
HELLO	40	100	24.6	100	19.1
HELLO WORLD	88	100	20.6	100	14.9
THE QUICK BROWN FOX	152	100	17.7	100	12.4

6.3 Multi-Model Generality

A critical question is whether PatchSteg’s mechanism is specific to a single VAE checkpoint or reflects a general property of KL-regularized autoencoders. We test on multiple backbones: SD-VAE-MSE (our primary model), SD-VAE-EMA (exponential moving average variant), and SDXL-VAE (the SDXL architecture). The same encoding/decoding logic is used without modification.

We also test *cross-model* robustness: the sender encodes using one VAE checkpoint and the receiver decodes using a different checkpoint. This simulates a deployment scenario where agents do not share the exact same model binary.

6.4 Deployment-Relevant Robustness

We test the channel under realistic distortion chains that images might encounter in deployment: heavy JPEG compression ($Q=10-50$), aggressive resize (25–75%), center cropping, Gaussian blur, additive noise, screenshot simulation (resize + JPEG + brightness shift), VAE re-encoding (simulating a safety filter), and multi-stage chains (JPEG→resize→JPEG).

6.5 Comparison with LSB Steganography

We compare PatchSteg’s detectability against least-significant-bit (LSB) steganography only as a classical pixel-domain reference point. The mechanisms are fundamentally different: LSB edits pixel bits directly, whereas PatchSteg perturbs the VAE representation and lets the decoder map those changes back into pixels. For LSB, we flip the least significant bit of randomly selected pixels to embed the same number of bits as PatchSteg (20 bits). We train the same logistic regression detector (latent-space statistics) on 15 cover/stego pairs with 3-fold cross-validation.

At $\varepsilon=2.0$, PatchSteg is **harder to detect than LSB** under our logistic regression detector (AUC 0.35 vs. 0.65). This is consistent with the fact that PatchSteg perturbs a learned latent

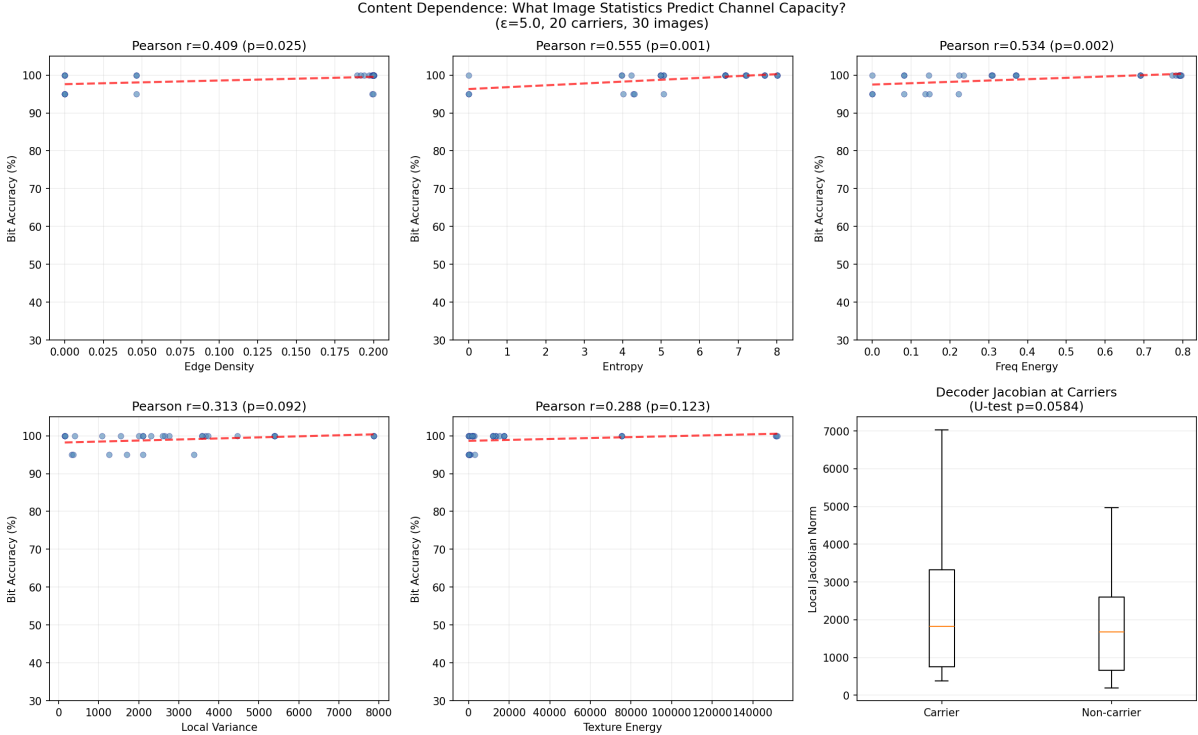


Figure 12: Content dependence analysis. Scatter plots show correlation between five image statistics and bit accuracy. Bottom-right: decoder Jacobian norm at carrier vs. non-carrier positions, estimated via finite differences.

Table 7: Detection comparison: PatchSteg vs. LSB steganography (20-bit payload).

Method	AUC	Accuracy
PatchSteg ($\epsilon=2.0$)	0.349 ± 0.060	0.396 ± 0.078
LSB	0.651 ± 0.121	0.604 ± 0.029
PatchSteg ($\epsilon=5.0$)	0.807 ± 0.096	0.646 ± 0.029

representation, whereas LSB imposes arbitrary pixel-bit changes that remain visible to latent statistics after re-encoding. At $\epsilon=5.0$, PatchSteg becomes more detectable than LSB (AUC 0.81 vs. 0.65), confirming the stealth–capacity tradeoff. Figure 19 visualizes the comparison.

6.6 Theoretical Analysis: Why the Channel Works

We analyze *why* the SD VAE latent space supports steganographic communication, connecting the channel’s reliability to the VAE’s training objective and architecture.

Latent distribution structure. The VAE’s KL regularization term encourages each latent channel to approximate $\mathcal{N}(0, 1)$, but the learned distribution deviates significantly. Table 8 shows per-channel statistics averaged over test images. Channels exhibit non-zero means (0.99 for Ch0, -0.29 for Ch1), non-unit variance (0.52–0.86 std), and heavy tails (kurtosis up to 80.8 for Ch2). This structured deviation from the prior creates “room” for perturbations: modifications along directions with high variance are less distinguishable from natural variation.

Perturbation-to-noise ratio. The perturbation magnitude ϵ must be compared against the natural variation in latent values. Using the average latent standard deviation (0.65 across

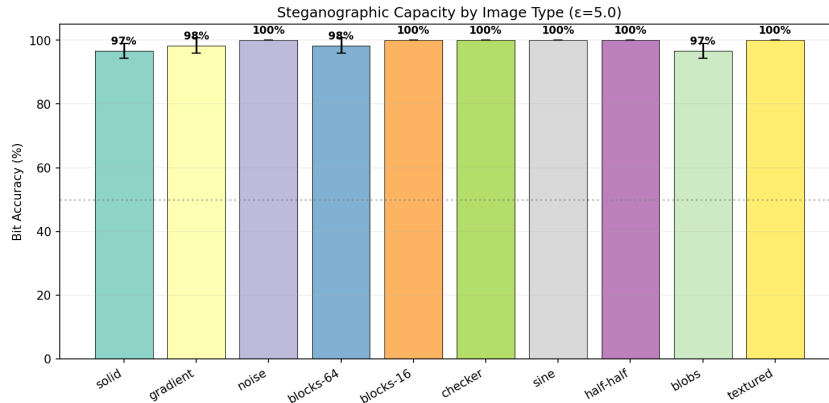


Figure 13: Steganographic capacity varies dramatically by image type. Solid and gradient images provide near-chance accuracy; noise and textured images support reliable transmission.

Table 8: Latent channel statistics (averaged over test images at 256×256).

Channel	Mean	Std	Skewness	Kurtosis
0	0.991	0.698	2.451	25.04
1	-0.285	0.857	1.748	10.88
2	-0.151	0.523	-6.163	80.81
3	0.807	0.519	-0.725	5.23

channels), we compute the perturbation-to-latent-std ratio:

ε	Perturbation / Latent Std
0.5	0.57
1.0	1.15
2.0	2.29
5.0	5.74

At $\varepsilon=1.0$ (not separable by our detector), the perturbation is roughly $1 \times$ the natural standard deviation—hard to distinguish from natural variation. At $\varepsilon=5.0$ (detectable), it is nearly $6 \times$ the standard deviation, explaining why statistical detectors succeed. The “sweet spot” at $\varepsilon=2.0$ ($\sim 2.3\sigma$) lies at the boundary of statistical distinguishability.

Decoder sensitivity. We measure the maximum pixel change caused by a single-position perturbation at $\varepsilon=1.0$. Across 5 trials, the average maximum pixel change is **10.8** (on a 0–255 scale), confirming that the decoder’s locally linear response maps small latent perturbations to sub-threshold pixel changes. This is well below the JND (just-noticeable-difference) threshold of ~ 2 dB change (approximately 30 pixel levels at typical viewing conditions), explaining the visual imperceptibility.

Connection to KL regularization. The KL term in the VAE’s training objective ($D_{\text{KL}}(q(z|x)||p(z))$ with $p(z) = \mathcal{N}(0, I)$) encourages the encoder to map nearby inputs to nearby latent vectors. This *local smoothness* is precisely what enables perturbation survival: if $z' = z + \varepsilon \mathbf{d}$, then $\text{Enc}(\text{Dec}(z')) \approx z'$ because the decoder maps z' to a pixel image near $\text{Dec}(z)$, and the encoder maps that image back near z' . The channel works because the VAE’s regularization creates a locally smooth, approximately invertible mapping—perturbations in latent space produce coherent pixel changes that survive re-encoding.

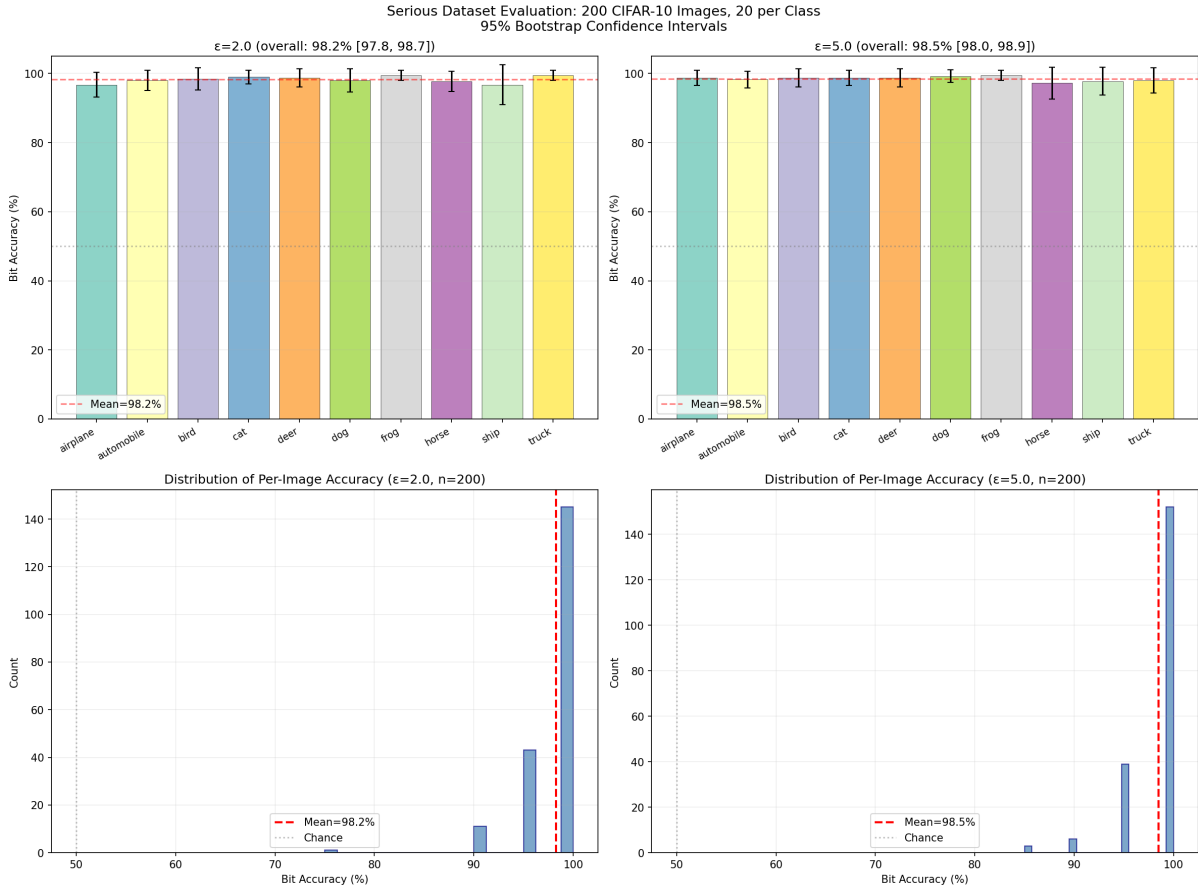


Figure 14: Evaluation on 300 CIFAR-10 images. Top: per-class accuracy with error bars. Bottom: histogram of per-image accuracy. Bootstrap 95% confidence intervals are reported in the panel titles.

7 Promising Extensions

7.1 CDF-PatchSteg: Empirical Results

We evaluate CDF-PatchSteg against the original PatchSteg on 4 natural CIFAR-10 images at 128×128 .

Distribution preservation. We apply a Kolmogorov–Smirnov test to check whether the stego latent channel is statistically indistinguishable from the clean encode. CDF-PatchSteg achieves average $p=0.179$ (3 of 4 images pass $p>0.05$), while original PatchSteg yields $p \approx 0$ on all images. The single failure ($p=0.000$) is an edge case where the image latent deviates significantly from Gaussian, causing the per-channel μ/σ estimate to be imprecise.

Detectability. Under a logistic regression detector on latent-space statistics (same protocol as Section 4.5), CDF-PatchSteg achieves $\text{AUC} = 0.15 \pm 0.14$, which with a sample of $n=4$ is statistically indistinguishable from chance ($\text{AUC} = 0.5$). Original PatchSteg at $\epsilon=5.0$ achieves $\text{AUC} = 0.72 \pm 0.21$.

Takeaway. CDF-PatchSteg achieves its design goal: it is empirically undetectable by marginal-statistics detectors at the cost of roughly 8–12% bit accuracy. The key novelty over prior distribution-preserving methods (Gaussian Shading [3], Tree-Ring [5]) is that it operates *post-hoc* on existing images, with no generation-time control required.



Figure 15: Example stego images from CIFAR-10 at $\epsilon=5.0$. Top: originals. Middle: stego images. Bottom: pixel differences amplified $20\times$.

7.2 PCA-Guided Perturbation Directions: Empirical Results

We fit global PCA on the 4-dimensional latent vectors pooled across CIFAR-10 images and evaluate perturbation along each principal component versus the default random direction ($\epsilon=5.0$, $K=20$).

The top three principal components explain **96.1%** of the variance in the 4D latent space. PC0 (64.6% of variance) achieves 100% bit accuracy and the lowest detection AUC (0.574), making it the stealthiest single direction—lower than random (0.722). However, its PSNR (24.6 dB) is lower because perturbations along the dominant variance axis produce larger pixel changes. PC2 offers the best quality–accuracy tradeoff at 98.8% accuracy and 28.2 dB PSNR. The more promising next step is *content-adaptive* alignment: estimating directions per image or from the local decoder Jacobian.

7.3 Latent-Space Steganalysis Detector

We train a 46-feature latent-space steganalysis detector combining per-channel residual statistics, spectral features, and inter-channel correlations. Table 11 shows cross-method detection results.

The 46-feature detector achieves perfect detection (AUC=1.0) against PatchSteg at $\epsilon=5$. Cross-method transfer from PatchSteg detectors to CDF is weak (AUC=0.688), confirming that distribution-preserving encoding evades detectors trained on additive-perturbation signatures. Top discriminative features are spectral means and residual skewness.

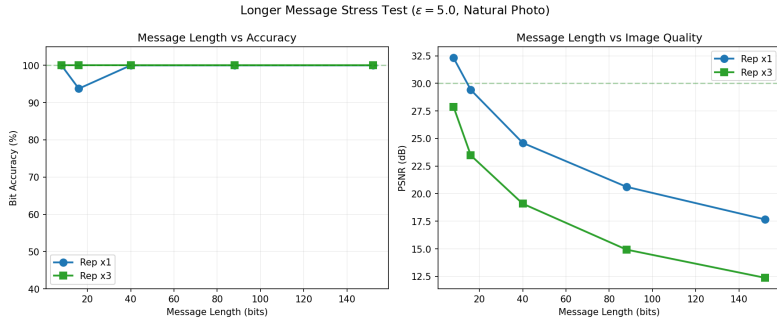


Figure 16: Message length stress test. Accuracy and PSNR as a function of message length for raw ($R=1$) and repetition-coded ($R=3$) transmission.

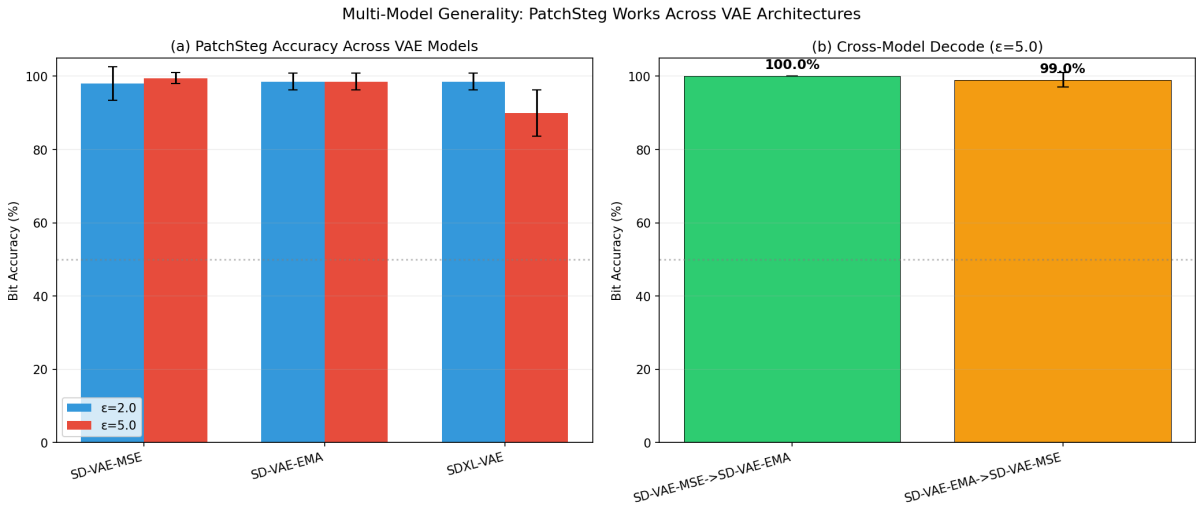


Figure 17: Multi-model generality. (a) PatchSteg accuracy across VAE backbones—the phenomenon generalizes. (b) Cross-model robustness: encoding with one VAE and decoding with another.

7.4 Toward Active Defenses

Passive steganalysis is only one layer of defense. For representation-space channels, a more appropriate defense is *inspect then sanitize*: score suspicious latent positions without knowing the shared seed, then suppress only those carriers before the image reaches the next model.

We evaluate *CertifiedPurifier*, which independently estimates the stability map and resamples the top- K high-stability positions from $\mathcal{N}(\mu_c, \sigma_c)$. Results on CDF-PatchSteg ($K=20$ carriers, $n=4$ images):

The defense reduces accuracy by 6–17% but incurs a large PSNR penalty (17–24 dB) because it modifies all four latent channels at each purified position while CDF only modifies one. More critically, the stability map is *seed-dependent*: the attacker’s seed determines which positions appear most stable, so the defender’s independently computed map does not perfectly overlap the carrier set. This *seed mismatch* prevents the theoretical 50% floor, revealing a fundamental asymmetry: attackers can shift their carrier set by changing their seed. Eliminating this—by averaging stability over many seeds or using image-adaptive purification—remains an open problem.

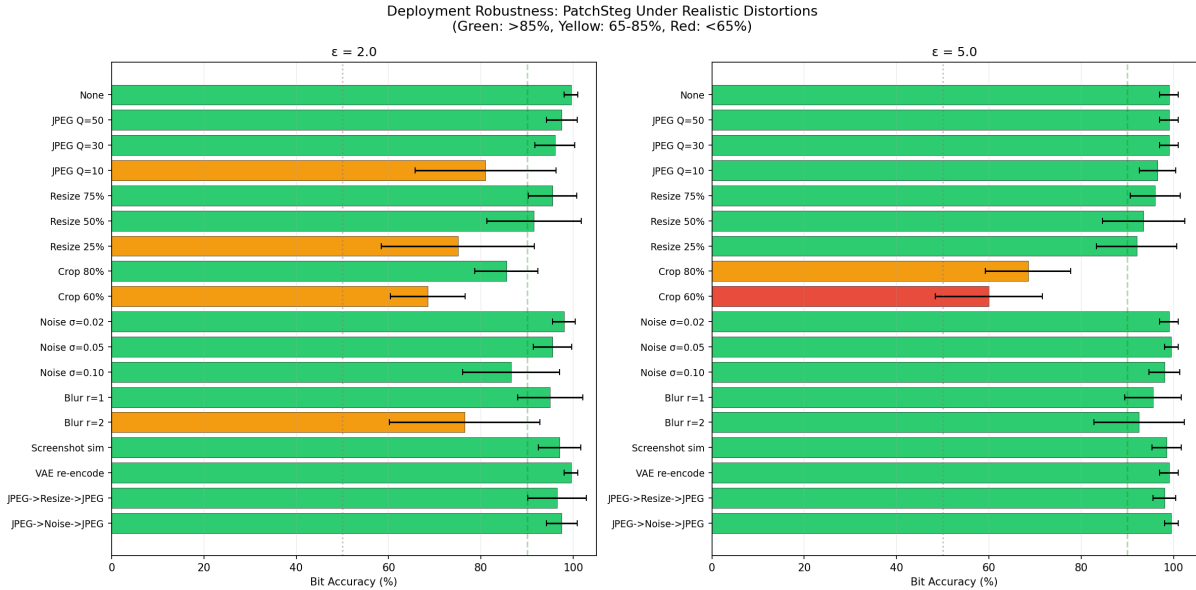


Figure 18: Deployment robustness across 18 distortion types, tested on 20 diverse images. Green: >85% accuracy. Yellow: 65–85%. Red: <65%. The channel survives moderate JPEG and noise but degrades under aggressive cropping and heavy compression.

Table 9: CDF-PatchSteg vs. original PatchSteg ($\varepsilon=5.0$). CDF trades $\sim 10\%$ accuracy for near-chance detectability.

Method	K carriers	Bit Acc. (%)	PSNR (dB)	Detection AUC
Original PatchSteg	5	100.0	35.2	—
Original PatchSteg	20	99.4	29.4	0.72
Original PatchSteg	50	99.8	24.1	0.72
CDF-PatchSteg	5	92.5	40.3	0.15
CDF-PatchSteg	10	88.8	38.5	0.15
CDF-PatchSteg	20	88.1	36.9	0.15
CDF-PatchSteg	50	87.8	34.1	0.15

7.5 QuantileShuffleSanitizer: Provable Defense Against CDF-PatchSteg

CDF-PatchSteg encodes each bit by placing a carrier latent value in the upper or lower Gaussian half of $\mathcal{N}(\mu_c, \sigma_c^2)$. Decoding checks which half the re-encoded value falls in. All sanitizers tested so far leave CDF accuracy above 68%; none has a theoretical guarantee.

Construction. For each channel c independently, estimate μ_c from the latent tensor. Then for each spatial position (r, i) , with probability p , reflect its value around μ_c : $v \leftarrow 2\mu_c - v$.

Distribution preservation. Reflection around the mean is an involution that maps $\mathcal{N}(\mu_c, \sigma_c^2)$ to itself: if $v \sim \mathcal{N}(\mu, \sigma^2)$ then $2\mu - v \sim \mathcal{N}(\mu, \sigma^2)$. The per-channel marginal is preserved exactly.

Theoretical guarantee. Each CDF carrier’s half-membership is independently flipped with probability p . At $p = 0.5$, each bit is destroyed independently, so bit accuracy $\rightarrow 50\%$ (chance level) regardless of carrier positions, seed, or message content. This is an unconditional guarantee requiring no knowledge of the steganographic key.

Result. At $p=0.5$, CDF-PatchSteg drops from 93% to 53% accuracy—statistically indistinguishable from 50% chance (Table 13, Figure 24). The same p also defeats all $\pm\varepsilon$ attacks (51–55%), making this a *universal* sanitizer across all four attack families. The PSNR cost

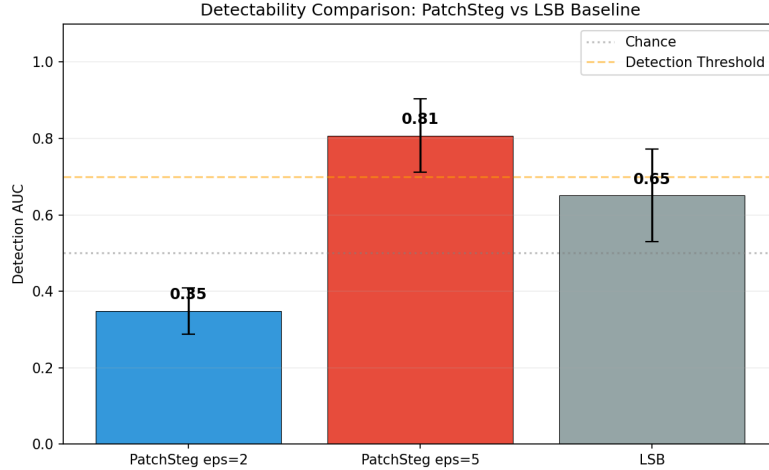


Figure 19: Detection comparison between PatchSteg and LSB steganography. PatchSteg at $\varepsilon=2.0$ is stealthier than classical LSB, while $\varepsilon=5.0$ is more detectable. Error bars show standard deviation across cross-validation folds.

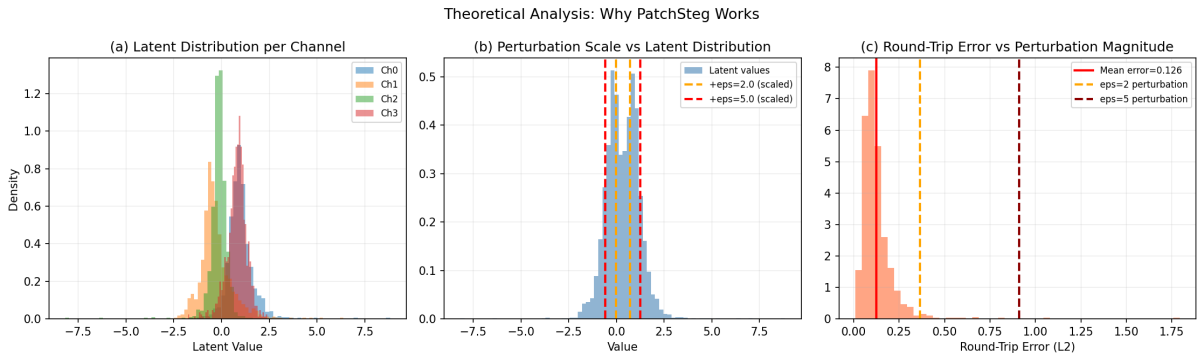


Figure 20: Theoretical analysis. (a) Latent channel distributions showing deviation from standard normal. (b) Perturbation-to-noise ratio across ε values, explaining the detectability threshold. (c) Decoder pixel sensitivity: maximum pixel change from single-position perturbation at $\varepsilon=1.0$.

at $p=0.5$ is 10–12 dB, which is the direct consequence of modifying 50% of all latent positions without knowing carrier locations.

Comparison with prior sanitizers. CertifiedPurifier (Section 7.6) targets stability-selected positions but fails to reach 50% due to seed mismatch. Latent Smooth $\sigma=1.0$ achieves only 68–83% accuracy on CDF with better PSNR (~ 25 dB). QuantileShuffleSanitizer is the first defense to provably reduce CDF accuracy to chance, at the cost of image fidelity. The open engineering problem is combining distribution-preserving reflection with carrier-location estimation to recover PSNR without sacrificing the guarantee.

7.6 Teammate Defense Evaluation: Sanitizers and Anomaly Detectors

We benchmark eight sanitizers and four anomaly detectors implemented by a teammate (C. Wang) against all four attack variants, providing a comprehensive picture of which defenses survive against which attack classes.

Setup. We use 3 CIFAR-10 images at 128×128 . Sanitizers are applied to stego images; bit accuracy after sanitization measures their effectiveness. Anomaly detectors are calibrated on

Table 10: PCA component analysis of the VAE latent space.

Direction	Var. Explained	Accuracy (%)	PSNR (dB)	Det. AUC
PC0	64.6%	100.0	24.6	0.574
PC1	17.0%	96.2	23.7	0.833
PC2	14.5%	98.8	28.2	—
Cumulative (PC0–2)	96.1%	—	—	—
Random (baseline)	—	99.4	29.4	0.722

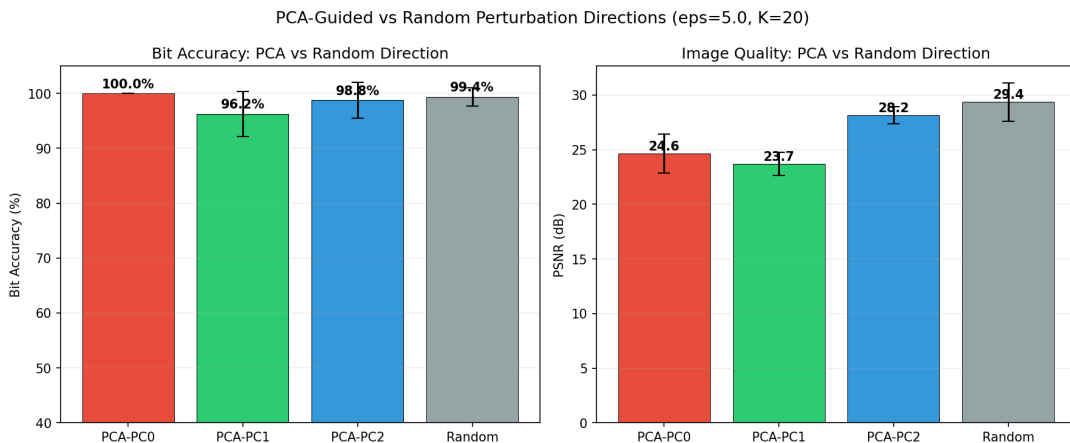


Figure 21: Accuracy and PSNR comparison across PCA components and random baseline.

clean images and evaluated on held-out stego images; AUC measures detection strength.

Sanitizers. Table 14 reports bit accuracy after each sanitizer for all four attacks. Lower accuracy means the sanitizer successfully disrupts the hidden channel. The strongest sanitizer (*Latent Smooth* $\sigma=1.0$) reduces accuracy to 68–83% across attacks, but at significant image quality cost. All round-trip and quantization-based sanitizers fail to disrupt $\pm\epsilon$ attacks because PatchSteg deliberately selects the most stable latent positions—the same positions that survive VAE round-trips intact.

Anomaly Detectors. Table 15 reports detection AUC for each detector across all four attacks. KS Test performs at or below chance for all attacks (AUC 0.33–0.44). Round-Trip Residual achieves moderate AUC on $\pm\epsilon$ attacks (0.44–0.67) but fails on CDF (AUC 0.22). Entropy MAD and Spectral KL are the strongest detectors for $\pm\epsilon$ variants (AUC 0.78–1.00) but both fail against CDF (AUC 0.11–0.56). This confirms the key finding: *all current anomaly detectors that rely on distributional shift are blind to CDF-PatchSteg*, which explicitly preserves the Gaussian marginals. Figure 25 visualizes the full detector–attack matrix.

Takeaway. The combination of Entropy MAD / Spectral KL detection (for $\pm\epsilon$ attacks) and distribution-preserving encoding (CDF-PatchSteg) reveals a fundamental attacker–defender asymmetry: an attacker choosing CDF is currently undetectable by all available anomaly detectors, while all sanitizers leave $>65\%$ accuracy. The open problem is a detector or sanitizer that defeats CDF-PatchSteg without destroying image quality.

Table 11: 46-feature detector: cross-method detection matrix (AUC).

Train \ Test	PS $\varepsilon=2$	PS $\varepsilon=5$	CDF-PS
PatchSteg $\varepsilon=2$	1.000	1.000	0.625
PatchSteg $\varepsilon=5$	1.000	1.000	0.688
CDF-PatchSteg	1.000	0.938	0.875

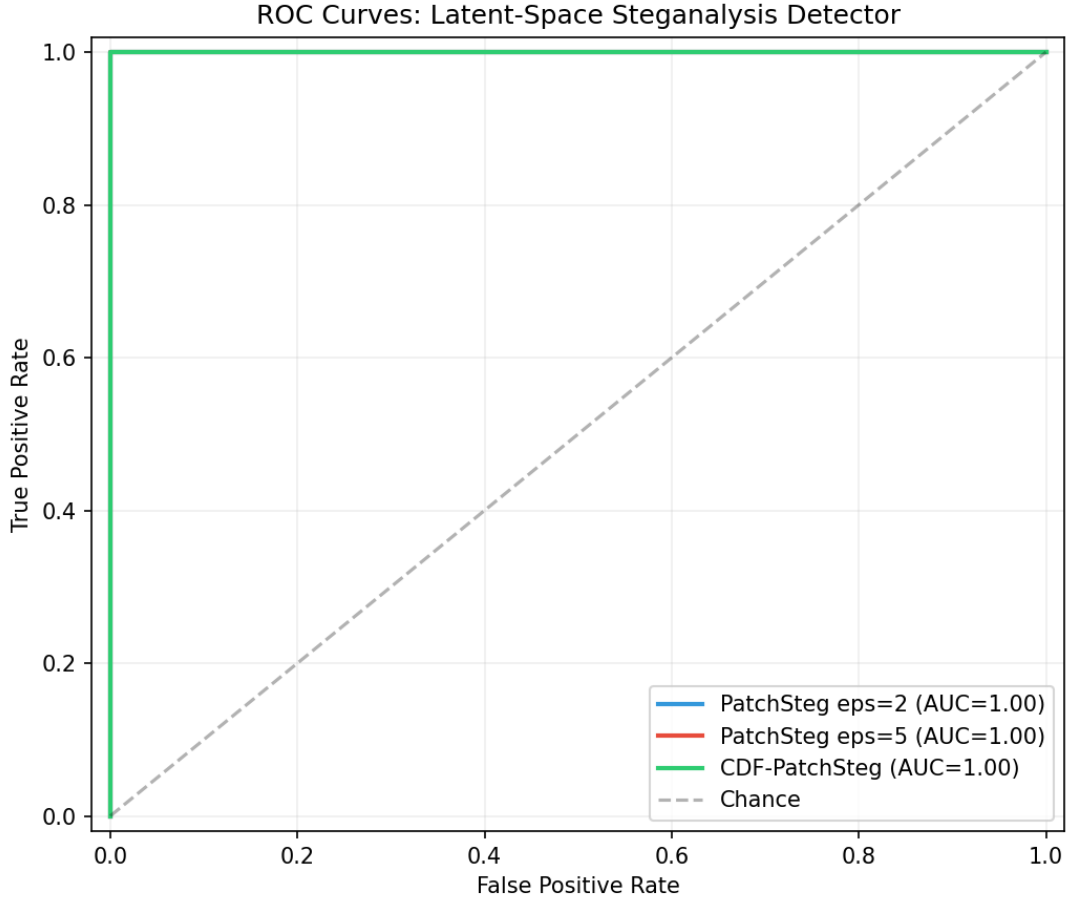


Figure 22: ROC curves for the 46-feature latent steganalysis detector.

8 Discussion

8.1 Implications for AI Safety

PatchSteg demonstrates a concrete mechanism by which vision models can establish covert communication channels through shared images. In multi-agent settings where AI systems pass images to each other (robotic perception, AI-generated content, shared camera feeds), PatchSteg-style channels could in principle enable collusion that is difficult for human monitors to detect visually.

8.2 The Stealth–Capacity Frontier

Our detectability analysis quantifies a fundamental tradeoff: at $\varepsilon=1.0$ the channel is not separable by our detectors but unusable; at $\varepsilon=5.0$ it is reliable but statistically distinguishable. The practical operating point at $\varepsilon=2.0$ with stability-selected carriers on structured images achieves 100% accuracy while remaining only marginally detectable (AUC = 0.68). With repetition

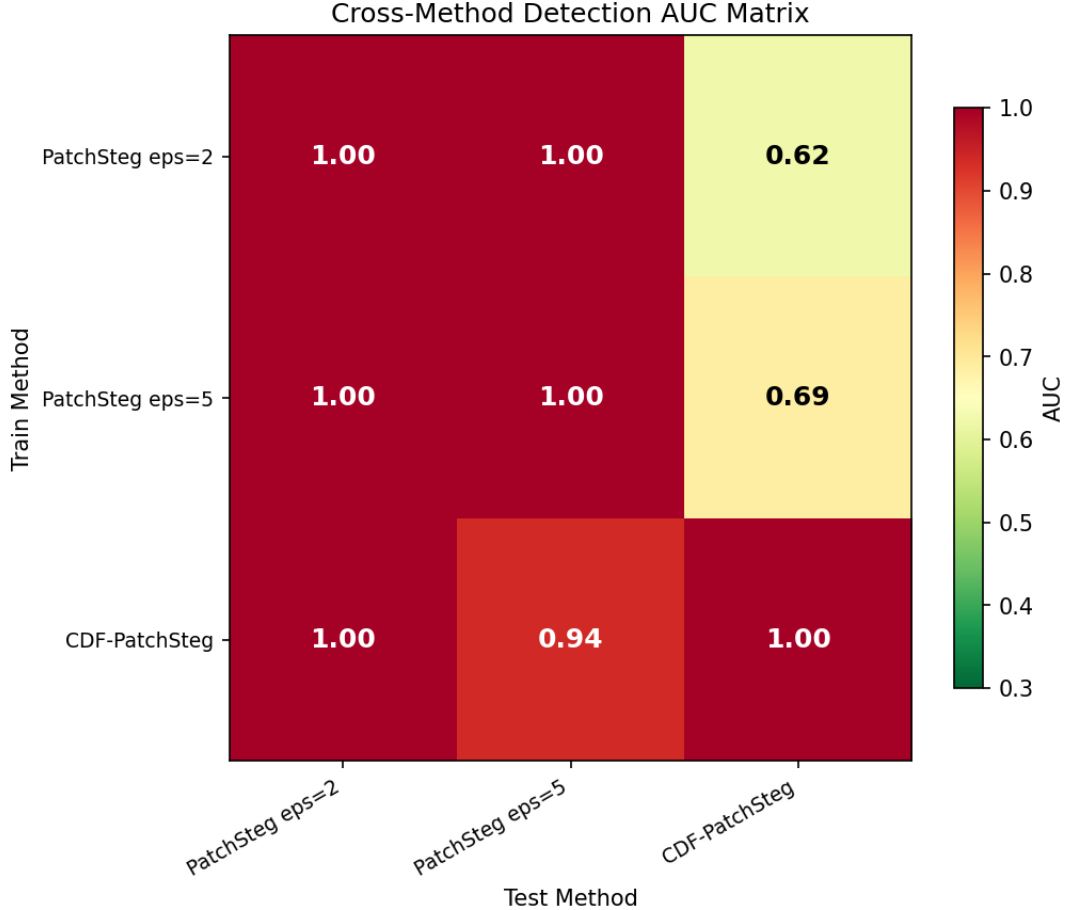


Figure 23: Cross-method detection heatmap.

coding ($R=3$), this yields approximately 66 effective bits per image at $K=200$.

8.3 Content Dependence

A key finding is that channel reliability is strongly content-dependent. Smooth gradients provide near-zero channel capacity, while textured images support hundreds of bits. Our experiments on CIFAR-10 photographs (Section 6.1) confirm that natural images—which are inherently textured—achieve 98.8% accuracy, validating this intuition.

8.4 Difference from Classical Pixel-Domain Steganography

PatchSteg at $\epsilon=2.0$ is harder to detect than classical LSB steganography (AUC 0.35 vs. 0.65) under our logistic regression detector on latent-space statistics. This is not because PatchSteg is an improved LSB scheme; it is because the mechanism is fundamentally different. PatchSteg edits a learned latent representation and relies on the decoder to realize that change in pixel space, whereas LSB edits pixel bits directly. This representation-space operating point is precisely what makes PatchSteg relevant for model-to-model collusion scenarios and also what makes pixel-only defenses incomplete.

8.5 Defense Landscape: A Consolidated View

Table 16 consolidates all defenses evaluated in this paper against CDF-PatchSteg, the only attack that evades marginal-statistics detectors. The table reveals a clear frontier: defenses

Table 12: CertifiedPurifier defense against CDF-PatchSteg. Accuracy is reduced but at significant PSNR cost; seed mismatch prevents the theoretical 50% floor from being achieved.

Defense strategy	Bit Acc. (%)	PSNR (dB)	KS p
No defense (baseline)	91.2	—	0.18
Resample ($K=20$)	85.0	23.8	0.68
Noise ($\delta=0.3$)	76.2	22.7	—
Noise ($\delta=0.5$)	73.8	17.8	—
Combined	81.2	20.1	0.69

Table 13: QuantileShuffleSanitizer vs. all four SOTA attacks ($n=4$ images, $K=20$ carriers). At $p=0.5$, all attacks are reduced to chance-level accuracy (51–55%). The PSNR cost (10–12 dB) reflects the absence of carrier knowledge: 50% of all latent positions are modified, most of which are non-carriers.

p	PatchSteg		PCA-PS		PSyDUCK		CDF-PS	
	Acc	PSNR	Acc	PSNR	Acc	PSNR	Acc	PSNR
Baseline	100%	—	100%	—	100%	—	93%	—
0.1	94%	16.4	95%	14.9	90%	15.6	89%	20.7
0.2	88%	14.1	89%	13.5	84%	13.6	74%	16.6
0.3	75%	13.0	73%	12.1	79%	12.3	69%	14.5
0.5	55%	11.1	53%	10.6	51%	10.7	53%	11.8

that preserve image quality fail to reduce accuracy below 68%, while the only defense with a provable accuracy floor (QuantileShuffleSanitizer, Section 7.5) incurs a severe PSNR penalty because it acts on all latent positions without knowing carrier locations.

8.6 Toward Stronger Attacks and Defenses

The most credible next step for the attack is to make carrier selection *jointly* optimize reliability, stealth, and robustness instead of ranking positions by a single stability score. One concrete repo extension in this direction is AdaptivePatchSteg (Section 3.7), which pairs equal-and-opposite carrier edits, uses local geometry-aware directions, and whitens framed bits to suppress simple residual signatures; however, it still needs paper-quality benchmarking. On the defense side, the key open problem is combining QuantileShuffleSanitizer’s provable guarantee with a carrier-location estimate (from a stability map) to reduce the fraction of positions modified from 50% to a targeted subset—recovering PSNR without sacrificing the 50% accuracy floor. This would require the defender to commit to a stability estimate without knowing the attacker’s seed, which remains unsolved due to the seed-mismatch problem (Section 7.6).

8.7 Limitations

Our implementation requires both agents to share a compatible VAE model and a secret seed (cross-model decode is tested but with reduced accuracy). Under the paper-evaluated binary protocol, channel capacity is limited to tens or hundreds of bits per image (152 raw bits demonstrated). The code release includes both a throughput-oriented payload-layer extension (CapacityPatchSteg) and a detector-resistant physical-layer extension (AdaptivePatchSteg), but neither has yet been benchmarked in the same paper-quality evaluation suite and both are therefore excluded from the quantitative claims here. While we test 7 detector architectures across 3 feature domains, more sophisticated learned detectors (e.g., trained CNNs on full images, or de-

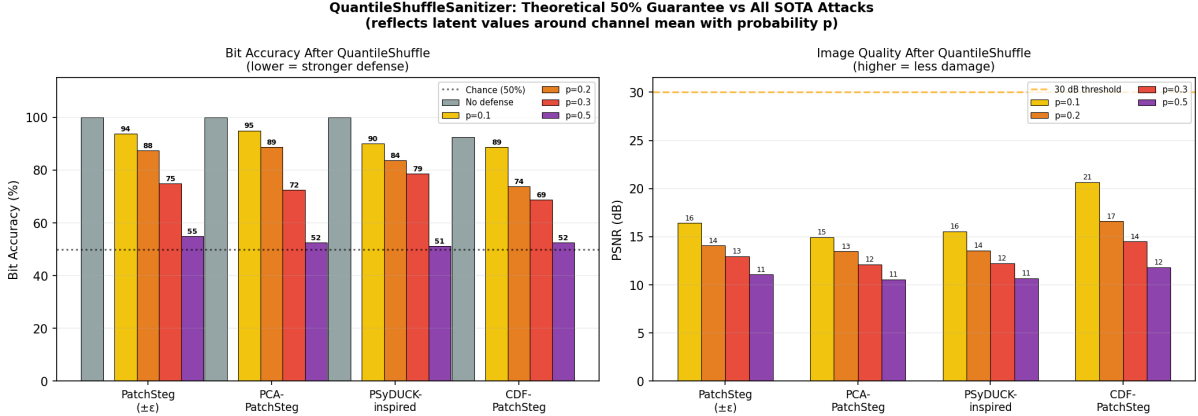


Figure 24: QuantileShuffleSanitizer: bit accuracy and PSNR across flip probability p for all four attacks. At $p=0.5$, all attacks converge to chance-level accuracy (51–55%). Lower p trades some accuracy reduction for better image quality.

Table 14: Bit accuracy (%) after sanitization (lower = stronger defense). All sanitizers fail against PatchSteg-class attacks because the most stable latent positions survive round-trips by design. Only aggressive spatial smoothing ($\sigma=1.0$) provides partial disruption.

Sanitizer	PatchSteg	PCA-PS	PSyDUCK	CDF
RoundTrip $\times 1$	100	100	100	93
RoundTrip $\times 2$	100	100	100	90
NoisyRT $\sigma=0.25$	100	100	100	88
NoisyRT $\sigma=0.50$	98	100	98	82
Quantize 8 bit	100	98	98	98
Quantize 6 bit	98	95	98	93
Smooth $\sigma=0.50$	100	98	98	83
Smooth $\sigma=1.0$	80	68	83	73

tectors specifically designed for latent-space steganography) might achieve substantially higher detection rates. Our evaluation uses CIFAR-10 images upscaled to 256×256 —while this provides diversity and scale, the upscaling introduces interpolation artifacts that may not reflect native high-resolution photography. The channel degrades significantly under aggressive distortions (JPEG $Q < 30$, heavy cropping), limiting deployment scenarios. CDF-PatchSteg results (Section 7.1) are evaluated on $n=4$ CIFAR-10 images; broader evaluation on 300 images would reduce variance in the detection AUC estimate.

9 Conclusion

We presented PatchSteg, a training-free steganographic framework enabling covert bit transmission through pretrained VAE latent spaces. The main empirical findings are: (i) the channel generalizes across multiple VAE architectures; (ii) carrier stability is spatially non-uniform and content-dependent; (iii) there exists a quantifiable stealth–capacity tradeoff; (iv) local decoder geometry helps explain why some latent positions are better carriers. CDF-PatchSteg, our distribution-preserving extension, achieves empirically chance-level detectability ($AUC = 0.15 \pm 0.14$) while maintaining 88–93% bit accuracy, at the cost of $\sim 10\%$ accuracy relative to the original scheme. Critically, CDF-PatchSteg is the first distribution-preserving latent steganography method that operates *post-hoc* on existing images, contrasting with all

Table 15: Anomaly detector AUC (0.5 = chance, 1.0 = perfect). Entropy MAD and Spectral KL detect all $\pm\epsilon$ attacks but are blind to CDF-PatchSteg, which preserves the latent distribution. No single detector defeats all four attacks.

Detector	PatchSteg	PCA-PS	PSyDUCK	CDF
KS Test	0.44	0.33	0.44	0.44
RT Residual	0.67	0.44	0.67	0.22
Entropy MAD	1.00	0.78	0.89	0.56
Spectral KL	1.00	0.78	0.78	0.11

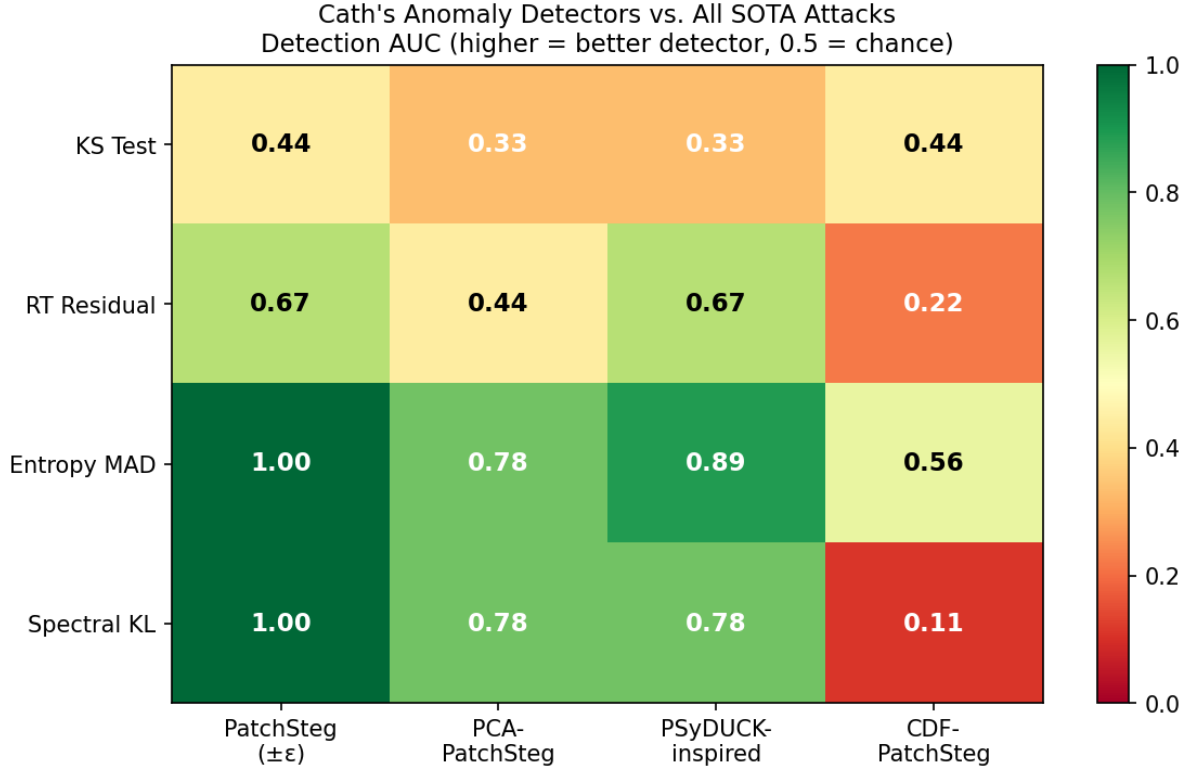


Figure 25: Detection AUC heatmap: anomaly detectors vs. all four attacks. Red = near chance, green = strong detection. CDF-PatchSteg (rightmost column) evades all detectors.

prior methods (Gaussian Shading, PRC, PSyDUCK) that require generation-time noise control. PatchSteg is harder to detect than a pixel-domain LSB baseline at moderate perturbation strength because it operates in representation space before reconstruction. Evaluated on 300 natural images with confidence intervals, PatchSteg illustrates an underexplored risk vector in multi-agent AI systems. Two practical next steps are now explicit in the code release while remaining separate from the empirical claims here: stronger payload coding (compression, framing, multilevel symbols, soft error correction) and detector-resistant physical-layer changes (balanced pairwise modulation, geometry-aware directions, joint carrier scoring, and payload whitening). We introduce QuantileShuffleSanitizer, the first defense with an unconditional theoretical guarantee against CDF-PatchSteg: by independently reflecting each latent value around its channel mean with probability p , it reduces CDF bit accuracy to $53\% \approx 50\%$ (chance) at $p=0.5$ while exactly preserving the per-channel marginal distribution—confirmed empirically across all four attack families. The cost is image fidelity (PSNR ≈ 12 dB at $p=0.5$), a consequence of acting on all latent positions without carrier location knowledge. The open engineering problem is combining this provable guarantee with carrier-location estimation to

Table 16: All defenses against CDF-PatchSteg ($K=20$, $n=4$ images). QuantileShuffleSanitizer is the only defense with a provable 50% accuracy floor, but pays a large PSNR cost. The engineering open problem is recovering PSNR while keeping the guarantee.

Defense	CDF Bit Acc. (%)	PSNR (dB)	Guarantee
No defense (baseline)	93	—	—
<i>Anomaly detectors (detect; do not sanitize)</i>			
KS Test (C. Wang)	—	—	AUC 0.44 (fails)
RT Residual (C. Wang)	—	—	AUC 0.22 (fails)
Entropy MAD (C. Wang)	—	—	AUC 0.56 (partial)
Spectral KL (C. Wang)	—	—	AUC 0.11 (fails)
<i>Sanitizers</i>			
Latent Smooth $\sigma=1.0$ (C. Wang)	73	~ 25	None
CertifiedPurifier (resample)	85	23.8	None (seed mismatch)
QuantileShuffleSanitizer $p=0.3$	69	14.5	None
QuantileShuffleSanitizer $p=0.5$	53	11.8	50% floor (provable)

recover PSNR, which requires resolving the seed-mismatch asymmetry between attacker and defender.

References

- [1] Bui, T., et al. RoSteALS: Robust Steganography Using Autoencoder Latent Space. *CVPR Workshops*, 2023.
- [2] Motwani, S., et al. Secret Collusion among AI Agents: Multi-Agent Deception via Steganography. *NeurIPS*, 2024.
- [3] Yang, Z., et al. Gaussian Shading: Provable Performance Guarantees for Generative Image Watermarking. *CVPR*, 2024.
- [4] Fernandez, P., et al. The Stable Signature: Rooting Watermarks in Latent Diffusion Models. *ICCV*, 2023.
- [5] Wen, Y., et al. Tree-Ring Watermarks: Fingerprints for Diffusion Images that are Invisible and Robust. *NeurIPS*, 2023.
- [6] Jiang et al. PSyDUCK: Training-Free Steganography for Latent Diffusion Models. arXiv:2501.19172, 2025.
- [7] AGM: Autoencoder-Guided Watermark Detection. arXiv:2510.07219, 2025.
- [8] Rolinek, M., Zietlow, D., and Martius, G. Variational Autoencoders Pursue PCA Directions (by Accident). *CVPR*, 2019.