ORIGINAL RESEARCH ARTICLE UHD JOURNAL OF SCIENCE AND TECHNOLOGY

# High-Capacity Video-in-Video Steganography using Residual Dense Blocks and CBAM Attention

**Brwa O. Abdalla, Omer Y. Abdulhammed**

*Department of Information Technology, Technical College of Informatics, Sulaimani Polytechnic University, Kurdistan Region, Iraq*

## ABSTRACT

Video steganography hides secret content inside video, but hiding a large payload while keeping the cover clean and the data secure remains difficult. This paper presents an encoder-decoder model that embeds a full RGB secret frame into a cover frame at 24 bits per pixel, frame by frame. The encoder joins the cover and secret into a six-channel input and learns a small additive residual, with Residual Dense Blocks and the Convolutional Block Attention Module preserving fine detail. On Vimeo-90K it reaches a stego PSNR of 45.68 dB and a recovery PSNR of 39.25 dB at a 0.78% bit error rate, results that hold on UCF-101, DAVIS 2017 and recorded video, with embedding at 24.8 frames per second on a single GPU. The stego video changes from frame to frame almost exactly as the cover does, and recovery is unaffected by lossless H.264 and H.265 coding. The stego frames pass six of seven classical statistical tests and are only partly flagged by StegExpose, while a robustness study finds good resistance to mild noise and blur but clear weakness against JPEG compression. The results indicate that residual dense learning with attention is an effective approach to full-frame video hiding.

**Index Terms:** Video Steganography, Data Hiding, Deep Learning, Residual Dense Block, Attention Mechanism, Steganalysis, Robustness

## 1. INTRODUCTION

Multimedia content, such as images, video, and audio, is now most used in communication, entertainment, education and healthcare [1]. The amount of multimedia that is shared over networks has escalated as continually internet and smart devices have expanded [2], and a lot of the information is shared via cloud platforms, creating significant concerns about the security and privacy of the data [3]. The most popular media type is video, with richer visual and temporal information than images and a significant portion of network traffic, including platforms, such as YouTube and Netflix [4]–[6]. The protection of video content is an interesting and relevant research field, since video is used in applications, such as telemedicine, video conferencing and surveillance, where security is important [1], [3].

Steganography hides secret information inside a normal digital file so that its presence is invisible to anyone not meant to see it [7]. Unlike cryptography, which makes data unreadable, steganography hides that any secret communication exists at all, so a third party sees only a normal media file [7], [8]. Video is an ideal cover: It holds far more information than a single image, its frames change gradually, leaving room to embed data without visible difference, and it is shared so

Access this article online

DOI:10.21928/uhdjst.v10n2y2026.pp198-216 E-ISSN: 2521-4217 P-ISSN: 2521-4209

Copyright © 2026 Abdalla and Abdulhammed. This is an open access article distributed under the Creative Commons Attribution Non-Commercial No Derivatives License 4.0 (CC BY-NC-ND 4.0)

**Corresponding author's e-mail:** Brwa O. Abdalla, Department of Information Technology, Technical College of Informatics, Sulaimani Polytechnic University, Sulaymaniyah, Iraq. E-mail: brwa.o.abdalla@spu.edu.iq

Received: 20-07-2026 Accepted: 11-10-2026 Published: 29-10-2026

widely that it is far less suspicious as a carrier than other file types [8], [9].

Traditional methods hide data by small manual changes to pixel values or frequency components, mainly least significant bit (LSB) substitution and wavelet or cosine transforms [7], [9]. These are simple but hide little data and leave traces that steganalysis can detect [7]. Deep learning is more effective, as an encoder learns to hide and a decoder to recover without manual design, giving better visual quality and more accurate recovery [8]. Even so, deep methods still face large model sizes, weak recovery, and poor resistance to real-world attacks, such as compression [7]–[9].

Effective video steganography must balance capacity, imperceptibility, recovery quality, and steganalysis resistance at the same time, yet full-frame encoder-decoder hiding at 24 bits per pixel (BPP) remains under-explored. Most deep methods work at much lower capacities, typically 0.1 to 4 BPP, to keep quality acceptable [10]–[12]. Some video-in-video methods reach 24 BPP, but they rely on 3D-CNN architectures [13] or temporal residual modeling [14] and neither combines residual dense blocks (RDBs) with attention. Invertible neural networks achieve high fidelity by modeling recovery as the inverse pass of the hiding network [7], [15], but neither work trains against distortions or reports a robustness evaluation. This motivates an encoder-decoder framework that embeds at the full 24 BPP while keeping high perceptual quality, and that can also be trained with a noise layer.

In this paper, we propose an encoder-decoder model that combines RDBs [16], convolutional block attention module (CBAM) attention [17], and multi-scale feature extraction. The dense blocks reuse hierarchical features to preserve fine detail, CBAM guides the network toward the most informative channels and regions, and multi-scale processing captures both local texture and broader context, which matters when the secret is a full frame rather than a binary message. The novelty lies in the combination rather than in any single component. PyraGAN [11] uses CBAM but carries binary messages at 1 BPP, VStegNET spreads the secret across groups of frames with 3D-CNNs at an average of 10.51 BPP, and the invertible methods train against no distortions. The proposed method places attention-guided RDBs and an optional noise layer in a single encoder-decoder trained at 24 BPP, so robustness can be traded against quality at full payload rather than being fixed by the architecture. Trained with a cosine loss schedule on Vimeo-90K [18], the model reaches a stego PSNR of 45.68 dB and stego Learned Perceptual Image Patch Similarity (LPIPS) [19] 0.0009, surpassing Weng *et al.* [14] by roughly 5 dB, and supports noise-layer training for better tolerance to moderate attacks, though high-strength JPEG compression remains a known limitation [20].

The objectives of this study are to design an encoder–decoder that hides a full-size secret frame at 24 BPP while keeping the stego frame perceptually indistinguishable from the cover, to recover the secret accurately from the stego frame alone, and to measure rather than claim the two properties that remain limited at this payload: The degree of tolerance to signal-processing distortion obtainable through noise-layer training, and the statistical detectability that persists at full capacity. The main contributions are: (1) An encoder-decoder model that combines RDBs, CBAM attention, and multi-scale feature extraction for full-frame video steganography at 24 BPP, with an optional noise layer for robustness; (2) cross-dataset evaluation on UCF-101, DAVIS 2017, and stock and self-recorded footage without retraining; (3) a security evaluation combining StegExpose [21] with seven further classical statistical tests, of which six showed no detection, with the Chi-square response examined in detail; and (4) near-real-time inference at 24.8 FPS embedding and 13.4 FPS extraction on a T4 GPU.

The rest of this paper is organized as follows: Section 2 reviews related work; Section 3 presents the proposed method; Section 4 the experimental setup; Section 5 the results and discussion; and Section 6 concludes and outlines future work.

## 2. RELATED WORK

In all the methods that have been reviewed, there is a clear trend that there is no single method capable of high capacity, good quality, robustness, and undetectability at the same time. Classical methods are simple but carry very little data and leave statistical traces that are detectable by steganalysis [22], [23]. Although deep image methods can enhance quality, they either perform much less than their full potential capacity [10], or report no robustness evaluation at all [15], or are valid only for still images [24], [25]. While hiding is extended to the temporal domain with deep video methods, there are also limitations, such as fragile frame handling [14], capacity below 24 BPP [11], [26], or evaluation confined to a single dataset with no robustness testing [26]. In this work, there are not many reports that combine robustness against real attacks, cross-dataset evaluation, and formal steganalysis.

The proposed method addresses part of these gaps in the encoder–decoder family: It supports hiding a full-size secret frame at 24 BPP, it is evaluated on three datasets and two further video pairs, it is evaluated with classical steganalysis, and it admits its limitations, such as compression robustness, openly.

### 2.1. Classical Steganography

In classical steganography, the secret information is hidden in the pixel values or the frequency domain (coefficients) of a cover. LSB substitution is the most popular spatial manipulation technique, which changes the LSBs of the cover pixels to secret bits, taking advantage of the fact that the eye is not sensitive to small intensity differences in the cover images [22], [25]. Simple yet hand-made rules that change the distribution of the cover, resulting in it being reliably detectable by steganalysis [10], [13]. The transform domain methods, such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) have better visual undetectability by putting in less perceptually sensitive frequency sub-bands [22], [23]. While these methods share two basic limitations, their embedding capacity is still rather low (usually considerably <0.5 BPP) [13], [25], and the hidden data are vulnerable to processing operations, such as video compression [23]. These constraints encouraged deep learning in which the encoder and decoder networks were able to encode and decode data end-to-end without having to design any manual feature [10], [25].

Ernawan [23] proposed a DWT-based scheme that selects embedding locations by scene-change detection, hiding data in the two-level DWT approximation coefficients of entropy-ranked 8×8 blocks. It reached an average stego PSNR of 46.09 dB and SSIM of 0.990 across seven videos, with strong robustness under MPEG-4 compression. However, it depends on object motion to trigger scene changes; its capacity is well below 1 BPP (binary text rather than structured media, so no recovery metric is reported), detectability was checked only by visual histogram comparison, and it was not tested against attacks, such as noise or blur.

Dalal and Juneja [22] embed a secret image only within the moving-object regions of frames, found by Gaussian Mixture Model foreground detection and blob tracking, on the assumption that changes in motion regions are less perceptible. Secret bits go into the middle-frequency sub-bands of a first-level 2D-DWT on those regions, after XOR-based encryption. Across eleven videos from CDNET and YouTube-8M, it reached an average stego PSNR above 45 dB with a bit error rate (BER) below 3.2% under salt-and-pepper noise, and detection by handcrafted steganalysis features peaked at 59.74%. Its capacity depends on the size and frequency of moving objects, making it unsuitable for low-motion content; robustness was tested against only one attack, and recovery quality was reported only as BER under noise, with no clean-condition metric for the extracted secret.

### 2.2. Deep Image Steganography

Deep image steganography replaces hand-crafted rules with encoder–decoder networks; four representative works are reviewed below.

Baluja [25] hides a full-size RGB secret inside a same-size cover using three networks (preparation, hiding, reveal) trained jointly, spreading the secret across all bits rather than the least significant ones. The payload is given as a 1:1 ratio rather than in BPP, which corresponds to the 24 BPP entry in Table 5, and resolution is not reported. On 1,000 ImageNet pairs the mean per-channel error was 3.0 for the cover and 3.2 for the recovered secret out of 256; PSNR and SSIM are not reported, so the values in Table 5 are taken from the re-evaluation in [27]. Robustness was not evaluated and the method is not intended for lossy formats, and while StegExpose was near random, CNN steganalyzers detected the containers at 90 to 95%.

Zhu *et al.* [10] proposed HiDDeN, an encoder–decoder trained with an adversarial discriminator on COCO images, with optional noise layers (dropout, cropout, crop, Gaussian blur, differentiable JPEG) inserted between them. Its steganography setting used no noise layer and hid short binary messages at 0.203 BPP in 512 × 512 BOSS images with bit error below $10^{-5}$, where the ATS steganalyzer detected it at 50% with unknown model weights but 98% with known weights. Robustness was evaluated only in a separate watermarking setting, so no robustness figure exists for the steganography model. SSIM and full-image hiding are not reported, and the 24 BPP and 28.19 dB entries in Table 5 come from the re-evaluation in [27].

**TABLE 1: Hiding and recovery quality of the base model on Vimeo-90K**

| Metric | Stego (cover vs. stego) | Recovered (secret vs. recovered) |
|---|---|---|
| PSNR (dB) | 45.68±2.28 | 39.25±2.44 |
| SSIM | 0.9848±0.0096 | 0.9665±0.0327 |
| LPIPS | 0.0009±0.0023 | 0.0099±0.0052 |
| BER (%) | NA | 0.78±0.63 |

LPIPS: Learned perceptual image patch similarity, BER: Bit error rate

Zhang *et al.* [24] proposed Universal Deep Hiding, in which a U-Net hiding network encodes only the secret into a cover-independent signal that is added directly to any natural cover, and a convolutional reveal network recovers it. It hides a full 128 × 128 RGB secret in a same-size cover, the full-image setting also used by Baluja. On ImageNet validation at 128 × 128, it reached a cover PSNR of 39.13 dB (SSIM 0.985), and a secret PSNR of 35.0 dB (SSIM 0.976), compared with 35.87 and 34.72 dB for a cover-dependent model the authors trained under the same settings. StegExpose failed to detect it but a trained DNN classifier did, which prior work attributes to the large capacity; no detection rate is reported. Robustness was evaluated against intensity shift, screen-to-camera capture and, in a watermarking setting, crop, cropout, dropout, blur and JPEG, where JPEG caused the largest secret error, and the method fails on covers with strong high-frequency noise. Video was not evaluated.

Jing *et al.* [15] proposed HiNet, an invertible neural network of 16 blocks in which concealing and revealing are the forward and backward passes of the same network and share its parameters. It hides a full-size secret image in a same-size cover in the wavelet domain, using a low-frequency wavelet loss to push most of the secret information into high-frequency sub-bands; no BPP figure is stated. Trained on DIV2K and tested on DIV2K (1024 × 1024), COCO, and ImageNet (256 × 256), it reached a stego PSNR of 48.99 dB (SSIM 0.9971) and a recovered-secret PSNR of 52.86 dB (SSIM 0.9992) on DIV2K. StegExpose gave an AUC of 0.5019 and SRNet 55.86% accuracy, although detection increased as SRNet was retrained with more samples. Robustness was not evaluated, and the authors note that concealing inevitably loses part of the secret information. All experiments used still images.

### 2.3. Deep Video Steganography

Video steganography is far more complex than image steganography, as it is required to deal with the temporal dimension, while maintaining imperceptibility for each frame. Early deep learning approaches followed the idea of frame-by-frame hiding from images, relying on the residual modeling between frames in a nearby temporal neighborhood to minimize the amount of residual to be hidden in each frame [14]. Later approaches relied on three-dimensional convolutional networks to more effectively model temporal interactions among frames [26], and required high model complexity, but were less flexible when dealing with different video content. Later, there were works that turned to invertible neural networks to enhance capacity and recovery, such as key-controlled recovery and scalable multi-video hiding [7]. Trained and tested only on Vimeo-90K, using 200 test sequences, it reached a stego PSNR of 45.17 dB (SSIM 0.980) and a recovered-secret PSNR of 48.39 dB for a single secret video, with no payload stated and no robustness evaluation. Despite this, deep video methods continue to have common limitations: Several methods make use of limited RGB-in-RGB capabilities [26], robustness to real world signal processing attacks is only sparsely examined in all the methods reviewed [7], [14], [26], and evaluation is typically limited to a single dataset and no cross-dataset validation. More recent work continues in these directions, including En-VStegNET, DRANet and Stego-STFAN for video, and ISN and Hashemi for images, all compared quantitatively in Tables 5 and 6.

Weng *et al.* [14] proposed what they describe as the first deep-network video steganography model, hiding a full-size color secret video in another, which corresponds to the 24 BPP entry in Table 5; no BPP figure is stated in the paper. An APD threshold splits secret frames into reference and residual frames, a residual being the difference from the latest reference frame, and two U-Net hiding networks, two reveal networks and a four-class classifier handle the two frame types. On a benchmark built from TRECVID MED 2017 at 256 × 256 it reached a container PSNR of 40.62 dB (SSIM 0.8466) and a revealed-secret PSNR of 40.76 dB (SSIM 0.8542) under lossless transmission. Robustness was evaluated against a single high-quality MP4 setting (FFMPEG, qscale 0, 100 Mbit/s), where the secret PSNR fell to 24.75 dB and a codec simulation layer raised it to 33.43 dB. The authors note that a misclassified frame affects subsequent frames until the next reference is correctly classified, although classification was fully accurate in their tests.

Mishra *et al.* [26] proposed VStegNET, in which 3D-CNN autoencoders based on the Hourglass micro-bottleneck architecture take 8 cover and 8 secret frames together and output 8 container frames, spreading the secret across frames rather than hiding it frame by frame, while the revealing network recovers the secret from the container frames alone. The authors report an average payload of 10.51 BPP without detailing the calculation. Using only UCF101 at 320 × 240, the model reached a mean container PSNR of 35.67 dB

**TABLE 2: Embedding and extraction efficiency on an NVIDIA T4 GPU (256×256 RGB frames)**

| Operation | Time per frame (ms) | Throughput (FPS) |
|---|---|---|
| Embedding | 40.37 | 24.8 |
| Extraction | 74.43 | 13.4 |

(SSIM 0.94) and a revealed-secret PSNR of 30.07 dB (SSIM 0.92). Robustness was not evaluated; the authors state that cropping, noise, or frame deletion can lose the secret and leaving this to future work. Detection accuracy was 53.3% for an SVM on SPAM features and 49.7% and 55.4% for pre-trained ResNet50 and Inception-v3, although classifiers trained end-to-end on labeled containers reached almost perfect detection, and quality degrades when frames are non-sequential.

Chai *et al.* [11] proposed PyraGAN, an end-to-end network that hides binary messages in VVC-compressed frames using a pyramid multi-scale encoder-decoder with a GAN discriminator and CBAM attention. On DIV2K and MSCOCO at 256 × 256, it reached a stego PSNR of 43.109 dB at 1 BPP with 99.4% extraction accuracy. However, it embeds only in I-frames, is tied to the VVC codec, carries only short binary messages at 1 BPP, and reports no robustness evaluation.

## 3. METHODOLOGY

### 3.1. Problem Formulation and Architecture Overview

Let a cover frame be C and a secret frame S, both RGB images of size 256 × 256 × 3 normalized to [0, 1]. The method works per frame: Each cover is paired with one secret and processed independently, and the stego video is formed by repeating this across all frames. The encoder E takes both frames and produces a stego frame G that looks almost identical to the cover; the decoder D takes only G and reconstructs an estimate S', never seeing the cover, mirroring a realistic setting where only the stego frame reaches the receiver.

Because each frame is handled on its own, a complete secret is carried in every frame rather than distributed across a group of frames, and performance does not depend on how similar neighboring frames are. Recovery also does not depend on frame order or on neighboring frames arriving intact, so a dropped or reordered frame affects only its own secret. Since the encoder adds only a small bounded residual, each stego frame stays close to its cover, so the cover's motion is largely carried through; the temporal consistency of the stego frames is evaluated in Section 5.7.

The cover and secret are joined along the channel dimension into a six-channel tensor (256 × 256 × 6), so the encoder reasons about both from its first layer. Rather than generating the stego frame directly, the encoder learns a small residual, bounded by a tanh and scaled by 0.1:

$$G = clip(C + tanh(E(C, S)) \times 0.1, 0, 1) \quad (1)$$

Where the final clipping keeps every pixel within the valid range [0, 1]. The recovery operation is written as:

$$S' = D(G) \quad (2)$$

Here the decoder ends in a sigmoid, which keeps every recovered pixel in [0, 1]. Because the secret is a full color image rather than a short bit string, hiding a 256 × 256 RGB secret in a same-size cover gives an embedding capacity of 24 BPP, the maximum for same-resolution color-into-color hiding and far above classical LSB payloads.

The proposed method is an encoder-decoder trained end-to-end. Two versions are reported: The Base model (Sections 5.1 to 5.4) and a Robust variant (Section 5.5), which continues training from the Base model with the noise layer added (Section 4.2). The Hiding and Revealing Networks are two separate networks, but both follow a similar U-Net-style design built from RDB and CBAM attention, detailed in Sections 3.2 and 3.3. The Revealing Network is larger because recovery is harder than hiding (about 20.37 million parameters against 8.76 million, roughly 29.13 million in total).

The overall architecture is summarized in Fig. 1, and the individual components are described in detail in Sections 3.2–3.5.

### 3.2. Hiding Network (Encoder)

The Hiding Network embeds the secret frame inside the cover. As outlined in Section 3.1, it has a U-Net-style design which includes a pre-processing stage, multi-scale feature extractor, contracting (downsampling) path, a bottleneck of RDB, and an expanding (upsampling) path that returns to full resolution. It does not produce the stego frame directly. Instead, it learns a small set of changes that hide the secret while keeping the cover almost unchanged.

The six-channel cover and secret input first pass through a pre-processing stage that expands it to a feature map that size is C × H × W along with 96 channels. A multi-scale feature extractor then looks into the input at several receptive-field sizes at once, which helps the network find good places to embed, since these depend on both fine local detail and broader context. The contracting path then downsamples the resolution in two stages while increasing the channel width (96→192→384), so each later layers can capture more global structure.

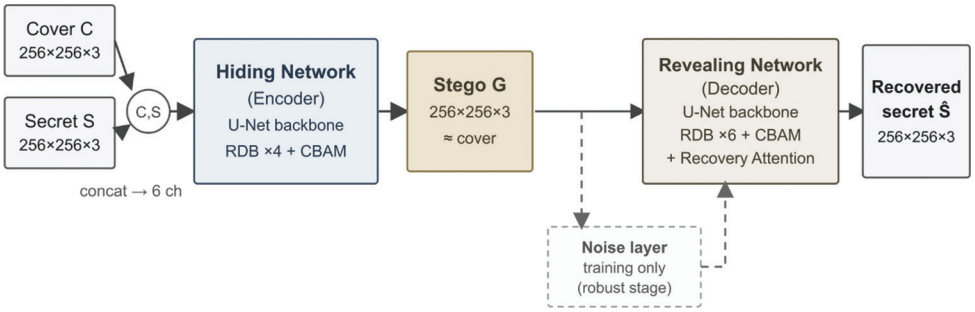



**Fig. 1.** Overview of the proposed method.

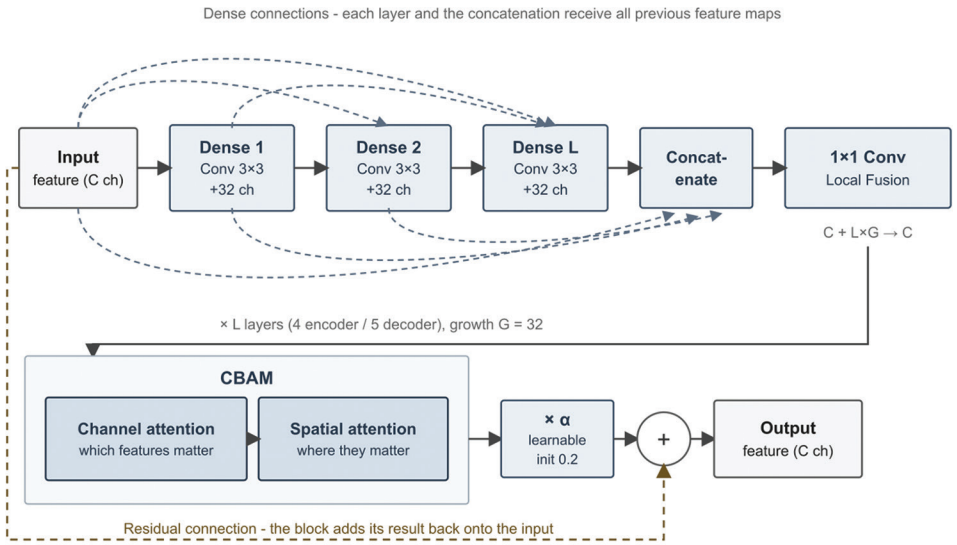



**Fig. 2.** The shared residual dense blocks with a convolutional block attention module block.

The bottleneck is a stack of four RDBs. Dense connections inside each block let every layer reuse the features of all earlier layers, which improves feature flow and keeps training stable. A CBAM is applied through the network to emphasize the most informative channels and spatial regions, which in steganography helps steer the embedding toward areas where changes are least visible.

The expanding path then upsamples back to the original resolution in two stages and reuses the earlier features to recover fine detail. A final layer maps the result to a three-channel output. This output is not the stego frame, but the small bounded residual that is added to the cover, as defined in Section 3.1. The RDB with CBAM used in this bottleneck is shared with the Revealing Network and is shown in Fig. 2.

Fig. 3 shows the Hiding Network. The six-channel cover–secret input passes through pre-processing to 96 channels, then multi-scale extraction, two downsampling stages (96 → 192 → 384), a bottleneck of four RDB with CBAM, and an expanding path with skip connections. The final layer outputs the three-channel bounded residual of Equation (1), not the stego frame itself.

### 3.3. Revealing Network (Decoder)

The Revealing Network will do the opposite; it must recover the secret from the stego frame without ever seeing the cover. Since the stego frame is nearly identical to the cover, the secret is hidden in very subtle perturbations, and recovery goes the harder route. It is therefore built on the same backbone as the Hiding Network (preprocessing, multi-scale feature extraction, a contracting path, a RDB bottleneck, and an expanding path with skip connections) but with higher capacity at each of these stages.

The network receives one stego frame with three channels; the cover and secret are now inside. The input is first expanded to 128 channels by two 3 × 3 convolutional layers, followed by

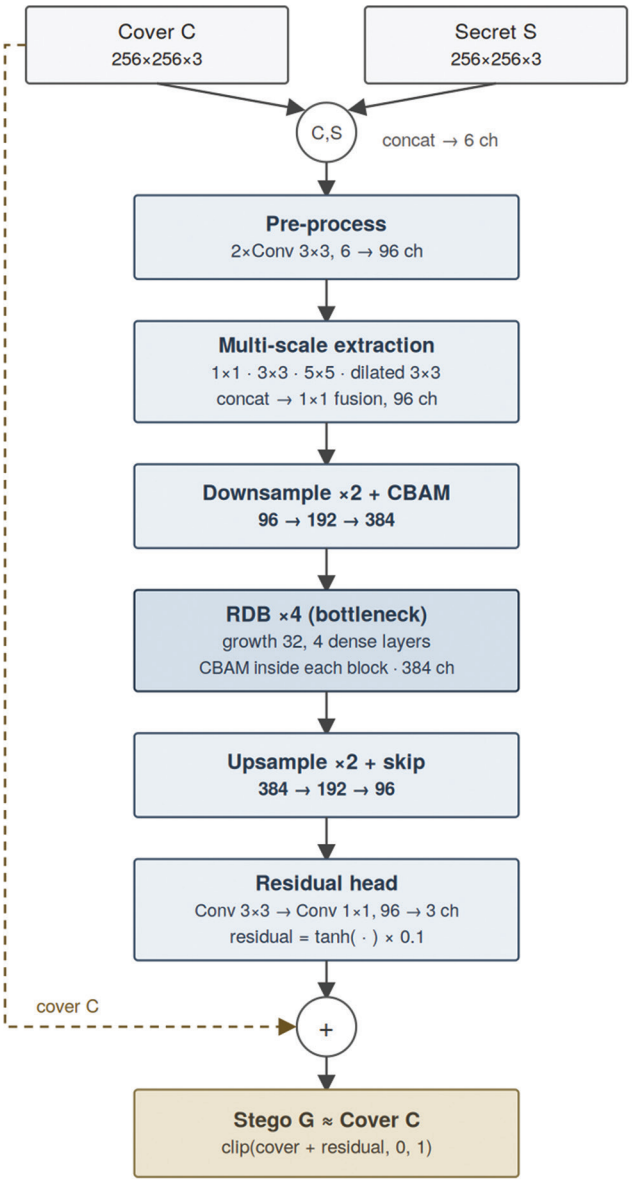



**Fig. 3.** The hiding Network (encoder).

the multi-scale extractor of Section 3.2. Two downsampling layers are added in the contracting path (128, 256, 512), followed by CBAM. The biggest difference between the bottleneck and the encoder is the number of blocks: 6 RDBs, each consisting of 5 dense layers, instead of 4 blocks of 4 dense layers. The operations of the RDB and CBAM are not repeated here, but are summarized in Section 3.2 (Fig. 2).

To enhance generalization, spatial dropout is used on the bottleneck features (rate 0.1, training only). This is then fed into a light-weight attention module dedicated to recovery that reweights channels and spatial regions to concentrate the decoder on those areas that most effectively encode the secret. It has a standard channel-and-spatial attention design and a slight bounded refinement to ensure the training is stable.

The expanding path mirrors the contracting path with two sub-pixel-convolution (pixel-shuffle) upsampling stages (512→256→128), concatenating the matching skip features at each stage to reconstruct sharp detail. A final 3 × 3 block and 1 × 1 convolution map the features to three channels. Unlike the encoder, which outputs only a residual, the decoder produces the recovered secret in full, with a sigmoid constraining S' to [0, 1].

Fig. 4 shows the Revealing Network, which mirrors the encoder at higher capacity. The three-channel stego frame is expanded to 128 channels, downsampled twice (128→256→512), and passed through six RDB of five dense layers each, with spatial dropout and a recovery-specific attention module at the bottleneck. Two pixel-shuffle upsampling stages restore full resolution, and a sigmoid constrains the recovered secret to [0, 1].

Fig. 2 shows the RDB with CBAM used in both networks. Each dense layer receives the outputs of all earlier layers, CBAM refines the fused result, and the block output is combined with its input through a learnable clamped scaling coefficient initialized to 0.2, which prevents runaway feature magnitudes during training.

### 3.4. Noise Layer Training

For the Robust variant only, a differentiable noise layer is inserted between the encoder and decoder during training, following the distortion-layer idea of HiDDeN. It applies one random distortion, such as approximate compression, additive noise, or blur, before recovery, so the decoder learns to recover from degraded frames. The distortions follow a curriculum that starts inactive and ramps up, so the network first masters clean hiding. The Base model is trained without this layer, and the layer is removed at inference; its effect is reported in Section 5.5.

### 3.5. Total Loss Function

The Hiding and Revealing networks are trained jointly by minimizing a single objective that balances two goals: Keeping the stego frame close to the cover and recovering the secret accurately. The total loss is the weighted sum of three reconstruction terms:

$$L_total = \lambda_cover\ L_cover + \lambda_secret\ L_secret + \lambda_ssim\ L_ssim \quad (3)$$

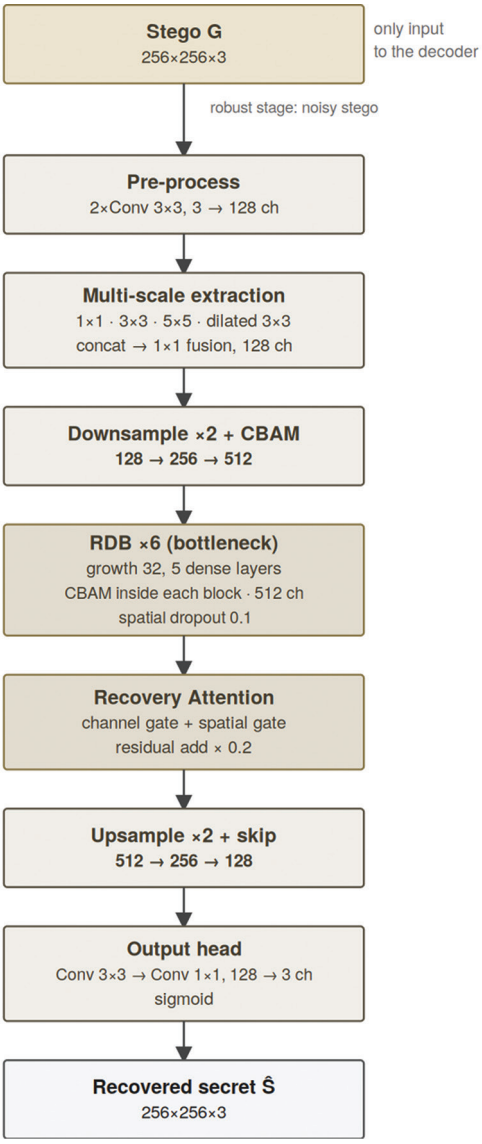



**Fig. 4.** The revealing Network (decoder).

L_cover is the cover fidelity loss, the mean absolute (L1) error between stego and cover, $L_cover = \| G - C \|_1$, which keeps the stego frame close to the cover and supports a high stego PSNR. L_secret is the secret recovery loss, the L1 error between the recovered and original secret, $L_secret = \| S' - S \|_1$. It drives accurate extraction and carries the largest reconstruction weight, since recovery is the harder direction; its weight is set several times higher than the cover weight. L_ssim is a structural loss based on the Structural Similarity Index, L_ssim = 1 − SSIM(G, C); unlike the pixel-wise L1 term, it encourages the stego frame to preserve the cover's structure, contrast, and luminance, which better matches human perception. The weights were chosen empirically. Preliminary runs showed that recovery is the harder of the two objectives, and that recovery quality stays poor unless λ_secret is set several times higher than λ_cover, so training begins with recovery dominant. The cosine schedule then shifts weight toward cover fidelity and structure as the secret becomes recoverable, with λ_cover rising from 0.4 to 0.9 and λ_ssim from 0.2 to 0.4 while λ_secret falls from 4.5 to 3.8. Stego quality is therefore refined only after recovery is established.

## 4. EXPERIMENTAL SETUP

### 4.1. Datasets and Pre-processing

The dataset used for training and validation was Vimeo-90K. From each clip, the first three frames were taken and resized to 256 × 256 using Lanczos resampling. The 9,597 available clips were shuffled once with a fixed seed and divided at clip level into 70/15/15 train, validation, and test splits, giving 6,717, 1,439, and 1,441 clips, respectively. Because the division is made over whole clips, no frame from a clip can appear in more than one split. Within each split, the clips were halved again, the first half providing cover frames and the second half secret frames, so cover and secret always originate from different clips and no clip serves as both. This yields 10,074 cover and 10,077 secret frames for training, 2,157 and 2,160 for validation, and 2,160 and 2,163 for testing.

Frames are stored as RGB PNG files named by split, role, clip index, and frame number, and are read in sorted filename order. Cover and secret are paired by index, so secret frame i is hidden inside cover frame i; the pairing is deterministic and identical in every epoch. The epoch length is set by the number of cover frames, so the three surplus secret frames in each split are unused. All frames are scaled to [0,1], and during training a random horizontal flip with $P = 0.5$ is applied, seeded identically for a cover and its paired secret so that both receive the same transform and stay aligned. No augmentation is applied to validation or test frames. To test generalization, two further datasets and two video pairs were used without retraining or fine-tuning: UCF-101, DAVIS 2017, clips from coverr.co, and a short self-recorded clip.

BER is computed directly on the pixel tensors rather than on an encoded bitstring. The original secret and the recovered secret are each thresholded at 0.5, the midpoint of the normalized intensity range, giving one binary value per color channel per pixel, and BER is the proportion of positions at which the two binary maps disagree. Each 256 × 256 RGB frame therefore contributes 3 × 256 × 256 = 196,608

comparisons. The same midpoint convention is used by Liu *et al.* [28] to binarize RGB secret images for bit-level evaluation of high-capacity deep hiding schemes.

Since thresholding records only which side of mid-intensity a value falls on, BER is reported together with PSNR, SSIM, and LPIPS, which measure the reconstruction error in full. Binarization conventions vary between studies, so BER is used here for comparison within this work rather than against values reported elsewhere.

### 4.2. Training Configuration

The full training configuration is summarized in Tables 3 and 4. Training ran in PyTorch on a single NVIDIA Tesla P100 GPU. The objective combined an L1 loss on the cover, an L1 loss on the recovered secret, and an SSIM loss between stego and cover, under a cosine curriculum that shifted emphasis from secret recovery toward cover fidelity and structure over the course of training.

The Base model was trained with the noise layer disabled throughout, with validation after every epoch, and the best validation checkpoint was kept, selected on BER with ties broken by recovery PSNR, provided BER stayed below 5%.

The Robust variant was obtained by continuing training from the Base model with the noise layer enabled. Before recovery, each batch was passed through one of a differentiable JPEG approximation, Gaussian noise, Gaussian blur, or salt-and-pepper noise, with a probability that scales with the curriculum strength, and the secret-recovery loss was weighted more heavily on attacked batches. All Robust variant results are reported from the checkpoint at which point the curriculum stood at 0.51 of full strength.

**TABLE 3: Base training configuration**

| Group | Item | Value |
|---|---|---|
| Data | Clips (train/val/test) | 6,717/1,439/1,441 |
| | Cover frames (train/val/test) | 10,074/2,157/2,160 |
| | Augmentation | Paired horizontal flip (train only), $P$=0.5 |
| Optimisation | Optimiser | Adam |
| | Learning rate | $2\times10^{-4}$, cosine annealing→$1\times10^{-6}$ |
| | Weight decay | $5\times10^{-5}$ |
| | Batch size | 12 |
| | Epochs | 310 |
| | Iterations per epoch | 840 |
| | Gradient clipping (L2 norm) | 1.0 |
| | Mixed precision | fp16 forward |
| | Random seed | 42 |
| Loss curriculum (cosine) | λ_cover (start→end) | 0.4→0.9 |
| | λ_secret (start→end) | 4.5→3.8 |
| | λ_ssim (start→end) | 0.2→0.4 |
| Hardware | GPU | NVIDIA Tesla P100 (16 GB) |
| | Framework | PyTorch |

**TABLE 4: Robust variant training configuration**

| Item | Value |
|---|---|
| Optimiser | AdamW |
| Learning rate | $8\times10^{-5}$ with cosine annealing |
| Batch size | 10 |
| Iterations per epoch | 1,008 |
| Distortion types | Differentiable JPEG approximation, Gaussian noise, Gaussian blur, salt-and-pepper |
| Noise curriculum | Linear increase in distortion strength; 0.51 of full strength at the selected checkpoint |
| Noise apply probability | 0.4, scaled by curriculum strength |
| Secret-recovery weight (attacked/clean) | 7.0/4.8 |

## 5. RESULTS AND DISCUSSION

Throughout this section, two versions of the proposed method are reported. The Base model is the clean model, trained without the noise layer, and is optimized for maximum imperceptibility and recovery quality. The Robust variant is trained with the robustness curriculum of Section 3.4 and trades clean-image quality (5.26 dB in stego PSNR) for improved tolerance to signal-processing distortion. Unless stated otherwise, the results below correspond to the Base model; the Robust variant is examined specifically in the robustness analysis (Section 5.5), where the trade-off between the two is quantified.

### 5.1. Hiding and Recovery Quality

This subsection reports the imperceptibility of the stego frame and the fidelity of the recovered secret on the Vimeo-90K test set (2,160 frames from 720 clips, Base model), using PSNR, SSIM [29], LPIPS, and BER.

The embedding is near-invisible. The stego frame reaches a PSNR of 45.68 dB, well above the 40 dB level usually regarded as imperceptible, with an SSIM of 0.9848 and an LPIPS of about 0.0009, so the cover and stego are almost identical in pixel, structural, and perceptual terms alike.

This is confirmed visually in Fig. 5: The cover and stego frames are indistinguishable to the eye, and the secret is recovered faithfully, while the amplified residual reveals the embedded secret only after ×10 magnification.

Beyond looking identical to the eye, the stego frame is also almost identical to the cover in its pixel values, and Fig. 6 shows this from several angles. Panels (a) and (b) show the cover and stego frames themselves; their grayscale statistics

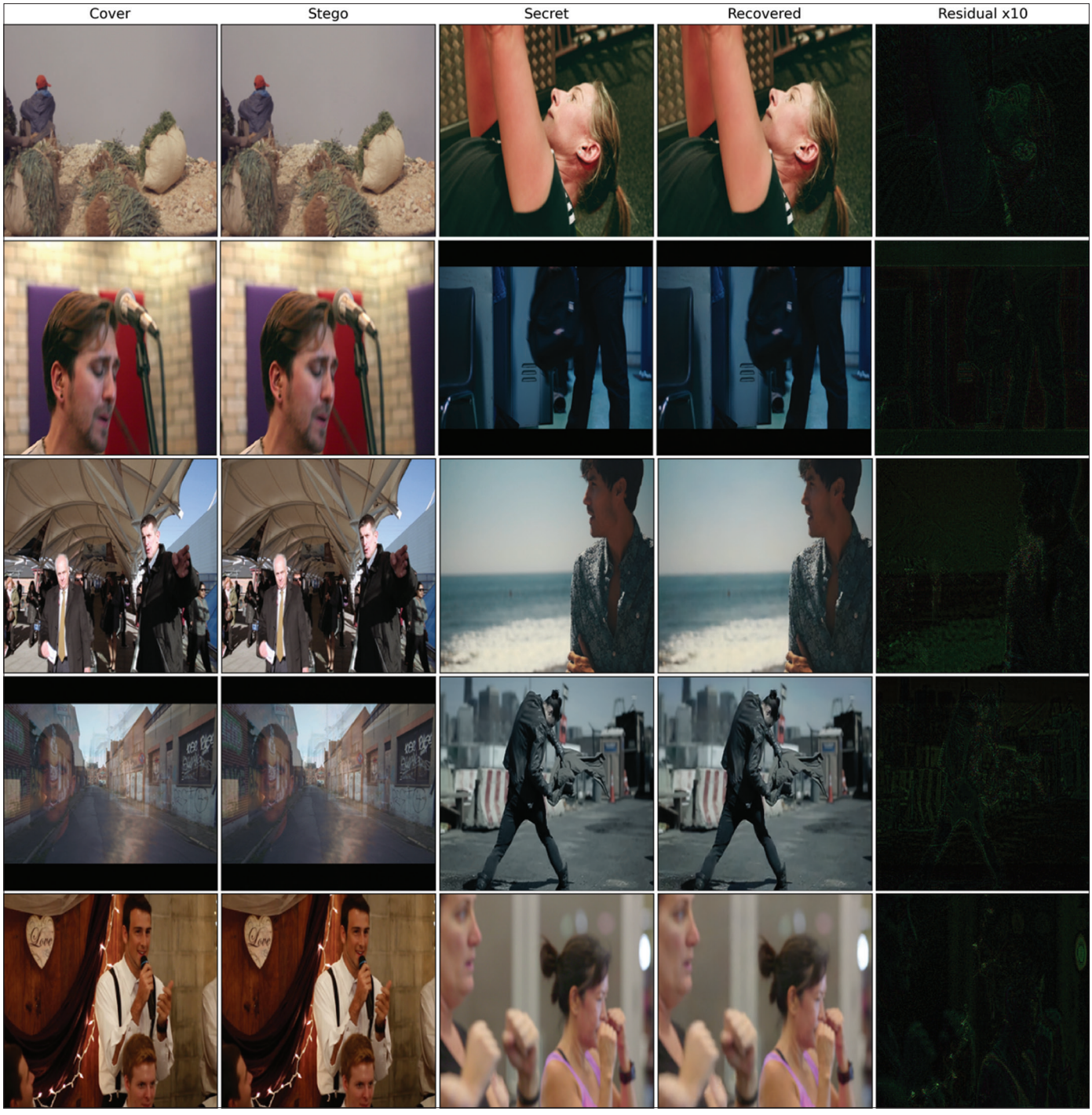



**Fig. 5.** Qualitative results on Vimeo-90K: Cover, stego, secret, recovered secret, and cover–stego residual (×10).

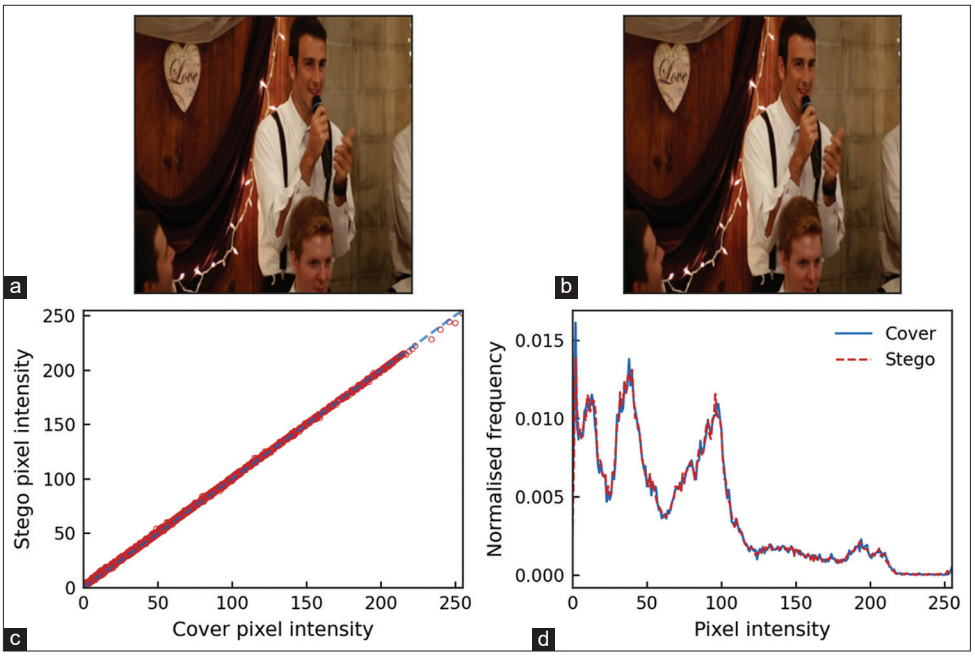



**Fig. 6.** Imperceptibility on a representative test frame: (a) cover frame; (b) stego frame; (c) cover-versus-stego pixel scatter; (d) cover and stego histograms overlaid.

share the same mean (67.7) and nearly the same spread ($\sigma$ = 50.9 vs. 50.7), so the embedding does not shift the frame's overall brightness or contrast. Panel (c) compares the frames pixel by pixel: The points fall almost exactly on the diagonal, so each stego pixel keeps nearly the same value as its cover pixel. Panel (d) places the two histograms on top of each other, where they overlap almost completely. Together these views show the hidden data is added so gently that the cover's appearance and pixel statistics stay almost untouched. The same analysis on the other four frames of Fig. 5 gives histogram intersections between 0.942 and 0.972 (mean 0.960), so this near-identical overlap is typical rather than a favorable case.

Recovery is the harder direction, yet remains strong. The recovered secret achieves a PSNR of 39.25 dB, an SSIM of 0.9665, and an LPIPS of 0.0099, while the BER measured over the 196,608 thresholded comparisons per frame is just 0.78%, meaning over 99% of the hidden information is restored correctly. Together these results show the system hides a full-size secret with negligible visible distortion while still reading it back accurately.

LPIPS is computed as a sum of squared feature distances [19], so it cannot be negative and is zero for identical images. Since the stego frames are almost identical to their covers, most LPIPS values lie close to zero, and a small number of frames with higher values give a right-skewed distribution. This is why the standard deviation (0.0023) is larger than the mean (0.0009). To check that these averages are representative, and not driven by a few strong frames, Figs. 7-9 examine the per-frame results across the whole test set.

Fig. 7 shows the distribution of per-frame stego PSNR. It forms a single tight peak, with the mean and median sitting almost on top of each other, so the reported average is not pulled up by a small group of outlying frames.

Fig. 8 ranks the same frames from best to worst. The curve falls only gently across the set, with no sharp drop at the lower end, and even the weakest frames remain near 38 dB.

Fig. 9 plots SSIM against PSNR for each frame. SSIM remains high across the whole PSNR range (median 0.987, with 95% of frames above 0.965), so frames with lower PSNR show only a small loss of structural similarity. Together, Figs. 7-9 show that the quality reported in Table 1 is consistent across the full test set.

The method is efficient when executed on a single GPU; embedding is 40.37 ms/frame (24.8 FPS) and extraction 74.43 ms/frame (13.4 FPS) on an NVIDIA T4 (as summarized in Table 2).

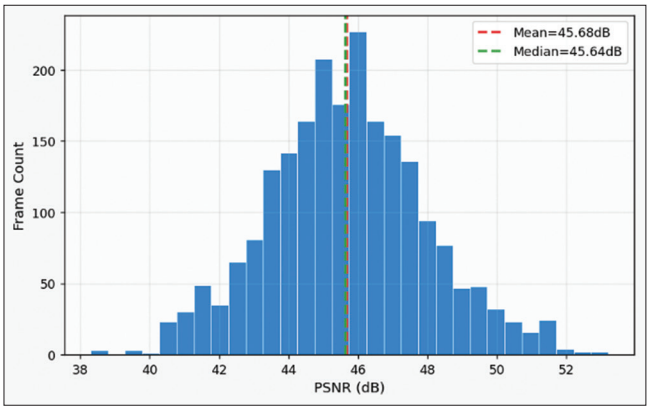



**Fig. 7.** Per-frame stego PSNR histogram on the Vimeo-90K test set.

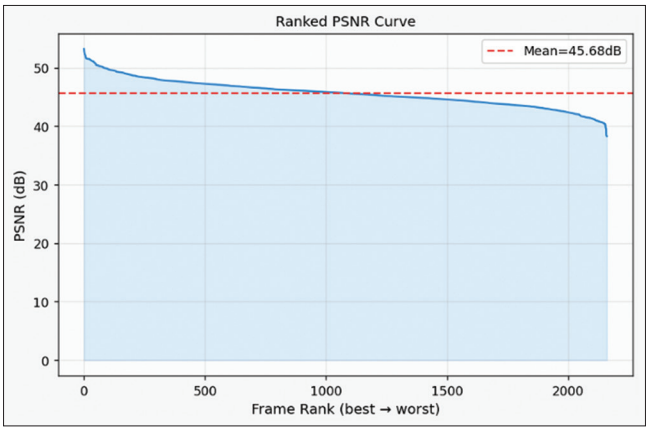



**Fig. 8.** Per-frame stego PSNR ranked best to worst.

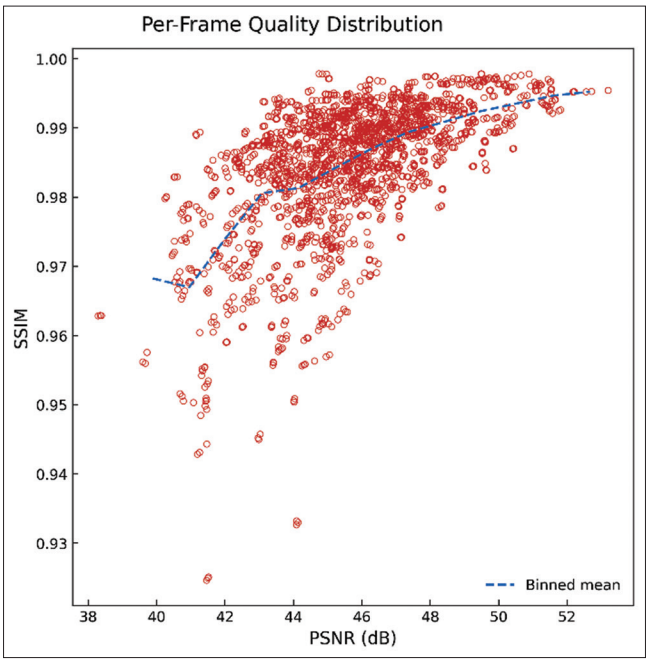



**Fig. 9.** Per-frame SSIM against stego PSNR.

## 5.2. Comparison with State-of-the-Art

Table 5 compares the proposed method against encoder-decoder baselines; INN methods follow a different architectural logic and are reported separately in Table 6. Among encoder-decoder methods, the proposed method achieves the highest stego PSNR (45.68 dB), exceeding the closest video baseline, Weng *et al.* [14] by roughly 5 dB at the same 24 BPP and resolution. Weng *et al.* do recover the secret at a slightly higher PSNR (40.76 vs. 39.25 dB), a trade-off between hiding and recovery quality. Baluja [25] and HiDDeN [10] also operate at 24 BPP but reach 28.42 dB and 28.19 dB stego PSNR, over 17 dB below the proposed method. Against VStegNET [26], the proposed method carries a heavier payload (24 vs. 10.51 BPP) while outperforming on all quality metrics; DRANet [30] is included with a limitation, as it was evaluated at 128 × 128 rather than 256 × 256. PyraGAN [11] embeds 1.0 BPP, where high stego quality is comparatively easier to achieve, so its 43.11 dB is not directly comparable with the 45.68 dB reported here at 24 BPP; the table should be read with capacity and quality together.

Baluja, HiDDeN and Hashemi are image methods, so this comparison is not directly equivalent: Their figures are per-image, whereas the values reported here are averaged over frames across video sequences. They are included because they operate within the same encoder-decoder paradigm at comparable payloads, and the Domain column separates the two groups; the same applies to ISN in Table 6.

Among the Invertible neural network INN-based methods in Table 6, LF-VSN reaches a stego PSNR of 45.17 dB, close to the proposed method, but recovers the secret at 48.39 dB, markedly higher than the 39.25 dB here. This gap reflects the invertible design, where recovery is the inverse pass of the hiding network rather than a separately trained decoder. ISN (38.05 dB stego) falls below the proposed method on hiding quality. Within the encoder-decoder paradigm, the proposed method therefore has the highest stego PSNR at 24 BPP. LPIPS is reported only here, as no existing baseline provides it.

Baluja [25] and HiDDeN [10] results are taken from the re-evaluation in [27], where both methods were retrained and evaluated under identical settings; HiDDeN's own work reports 0.203 BPP for a bit-message configuration.

LF-VSN [7] does not state a payload. The 24 BPP entry is inferred, since it hides a secret video of the same size as the cover. The values shown are for one secret video,

**TABLE 5: Quantitative comparison with encoder–decoder steganography methods**

| Method | Domain | BPP | Stego PSNR (dB) | Rec. PSNR (dB) | Stego SSIM | LPIPS (stego) | BER (%) | Params (M) |
|---|---|---|---|---|---|---|---|---|
| Baluja [25] | Image | 24 | 28.42 | 28.53 | 0.93 | N/R | N/R | N/R |
| HiDDeN [10] | Image | 24 | 28.19 | 28.42 | 0.92 | N/R | 0.001 | ~0.45 |
| Weng *et al.* [14] | Video | 24 | 40.62 | 40.76 | 0.846 | N/R | N/R | 314.16 |
| Hashemi *et al.* [41] | Image | 8 | 39.06 | 35.81 | 0.98 | N/R | N/R | N/R |
| Stego-STFAN [42] | Video | 24 | 27.03 | 23.09 | N/R | N/R | N/R | 43.91 |
| VStegNET [26] | Video | 10.51 | 35.67 | 30.07 | 0.94 | N/R | N/R | N/R |
| En-VStegNET [13] | Video | 24 | 36.62 | 32.88 | 0.95 | N/R | N/R | N/R |
| DRANet [30] | Video | 24 | 31.11 | 28.45 | 0.97 | N/R | N/R | 16.54 |
| PyraGAN [11] | Video | 1.0 | 43.11 | N/R | N/R | N/R | 0.6 | ~0.36 |
| Proposed method (base) | Video | 24 | 45.68 | 39.25 | 0.9848 | 0.0009 | 0.78 | 29.13 |

N/R: Not reported, LPIPS: Learned perceptual image patch similarity, BER: Bit error rate

**TABLE 6: Quantitative comparison with INN-based video steganography methods**

| Method | Domain | BPP | Stego PSNR (dB) | Rec. PSNR (dB) | Stego SSIM |
|---|---|---|---|---|---|
| ISN [43] | Image | 24 | 38.05 | 35.38 | 0.954 |
| LF-VSN [7] | Video | 24 | 45.17 | 48.39 | 0.980 |
| Proposed method (Base) | Video | 24 | 45.68 | 39.25 | 0.9848 |

which is its best case; quality falls as more secret videos are added.

### 5.3. Cross-Dataset Generalization

To test generalization, the Base model trained only on Vimeo-90K was evaluated without retraining or fine-tuning on four additional sources: UCF-101, DAVIS 2017, clips from coverr.co, and a recording made by the author.

For UCF-101, action classes were divided into disjoint cover and secret sets so that cover and secret frames always come from different action categories. Videos within each class were then partitioned into training, validation, and test portions, and only the test portion was used here; no UCF-101 data were used at any stage of training. Up to four evenly spaced frames were extracted from each clip, giving 2,790 frames from 699 clips. DAVIS 2017 was evaluated on its validation set: 30 sequences partitioned into 15 disjoint cover clips and 15 secret clips, so no sequence is hidden within itself. All 1,999 validation frames were used, giving 1,006 cover–secret pairs; because the two halves contain 1,006 and 993 frames, 13 secret frames appear in two pairs. The coverr.co material consists of two freely licensed stock clips at 1920 × 1080 and 24 frames per second: A moving camera along a mountain road under a clear sky with snow and clouds as the cover, and a pedestrian walking toward the camera on a street with overhead power lines, trees, and a wall as the secret, giving 365 frame pairs. The author's recording consists of two clips at 1024 × 576 and 30 frames per second: A busy urban street as the cover, and an indoor office scene containing furniture, a computer, and office equipment as the secret, giving 122 frame pairs. All frames were resized to 256 × 256 and processed exactly as in training.

These sources differ from Vimeo-90K and from each other in content, motion, and recording conditions. The results are summarized in Table 7. Stego PSNR stays above 43 dB across the three datasets and the two video pairs, with a spread of 2.27 dB between the highest (Vimeo-90K, 45.68 dB) and lowest (DAVIS 2017, 43.41 dB). Recovery PSNR varies more, from 36.41 dB on DAVIS 2017 to 39.25 dB on Vimeo-90K. BER stays below 2% across all datasets.

These results indicate that performance is stable on footage outside the training distribution. The UCF-101 result carries the most weight, since it averages over 2,790 frames from 699 clips that were never seen in training. Together with Vimeo-90K, this covers 1,419 clips of varied content and motion. The DAVIS 2017 result adds a second multi-clip dataset, evaluated over 1,006 pairs, and gives the lowest figures of all test sources, remaining within 2.3 dB of the training-domain stego result. The coverr.co and author-recorded results each come from one clip pair and show that the method continues to work on footage very different from either dataset.

The model was also tested outside its training resolution. The author's recording was re-processed at 512 × 512 using the same Base model without modification, giving 45.18 dB stego and 38.07 dB recovery at 0.64% BER, against 44.34 dB and 38.04 dB at 0.76% BER for the same clips at 256 × 256. The embedding rate is unchanged at 24 BPP, since cover and secret scale together. This is a single clip pair of 122 frames

**TABLE 7: Cross-dataset generalisation of the proposed method**

| Dataset | # Clip pairs | # of Frames | Stego PSNR (dB) | Rec. PSNR (dB) | Stego SSIM | LPIPS (stego) | BER (%) |
|---|---|---|---|---|---|---|---|
| Vimeo-90K | 720 | 2160 | 45.68 | 39.25 | 0.9848 | 0.0009 | 0.78 |
| UCF-101 | 699 | 2790 | 44.27 | 37.51 | 0.9803 | 0.0007 | 1.02 |
| DAVIS 2017 | 15 | 1006 | 43.41 | 36.41 | 0.9852 | 0.0004 | 1.80 |
| coverr.co | 1 | 365 | 44.16 | 38.09 | 0.9883 | 0.0003 | 1.31 |
| Author's recording | 1 | 122 | 44.34 | 38.04 | 0.9846 | 0.0004 | 0.76 |

LPIPS: Learned perceptual image patch similarity, BER: Bit error rate

and is reported as a demonstration that the network operates at a resolution not seen in training, not as a characterization of its behavior across resolutions.

### 5.4. Security Analysis (Steganalysis Resistance)

Evaluating steganographic security requires detectors that had no role in training. Security is therefore assessed with StegExpose, a widely used classical tool that fuses several spatial-domain detectors into a single score, which makes it a genuinely independent benchmark.

The evaluation used 200 cover and 200 stego frames from the Vimeo-90K test set, embedded at 45.73 dB on this 200-frame subset. StegExpose assigns each frame a fusion score, the mean of its spatial-domain detectors, which for this run were Primary Sets, Sample Pairs, and RS analysis. All figures below refer to this fusion score, and 0.2 is the tool's default operating threshold, used here so that the reported rates are comparable with published results using the same tool. At that threshold, StegExpose flagged none of the 200 cover frames (FPR = 0.00) and only 46 of the 200 stego frames (TPR = 0.23), an accuracy of 61.5%, only modestly above random guessing. Mean scores were $0.030 \pm 0.032$ for cover and $0.115 \pm 0.141$ for stego; these are reported alongside the rates because the threshold sets the balance between false positives and missed detections, so the result should not rest on that single choice. The two distributions also overlap substantially. The highest fusion score among the 200 cover frames was 0.167, and 148 of the 200 stego frames scored at or below that value, meaning that three quarters of the stego frames were no more suspicious to the tool than an ordinary cover frame. This comparison does not depend on the threshold, and indicates that detection is confined to a small group of outlier frames rather than a systematic difference between the two sets. This matches other deep hiding methods at 24 BPP [24], [25], where StegExpose also performs poorly; in those works this is attributed to deep networks embedding in high-frequency components rather than in the LSB these tests target. The partial detection (TPR = 0.23) reflects some statistical deviation at full capacity, and is reported openly rather than overstated as complete undetectability.

Seven further classical tests on 500 cover and 500 stego frames (mean–variance, histogram difference, chi-square, entropy, pixel correlation, RS analysis, and LSB distribution) found nothing in six cases; only chi-square flagged the stego frames. This is not a sign of LSB hiding: The LSB-ratio test stays clear ($\Delta = 0.010$), and the stego Chi-square value is far below the cover's (198,444 against 717,376), the reverse of what LSB hiding would cause. The histogram in Fig. 6 agrees: Cover and stego almost fully overlap (intersection 0.97), and Chi-square flags the cover frame too, so it cannot tell them apart. These results show no classical steganalysis trace other than the chi-square response, which does not separate cover from stego.

These results characterize resistance to classical statistical steganalysis only. Learned steganalysis detectors operate on features that the tools used here do not model, and no claim is made that the method resists them at this payload. The security evaluation reported in this section should therefore be read as evidence that the method leaves no classical LSB-type trace, not as evidence of steganographic security in general.

### 5.5. Robustness Analysis

A per-attack evaluation was carried out on the Vimeo-90K test set (2,160 frames) using both the Base model and the Robust variant, covering twelve attack configurations across four types: JPEG compression (Q = 95, 85, 70), Gaussian noise ($\sigma = 0.005, 0.01, 0.02$), Gaussian blur ($\sigma = 0.3, 0.5, 1.0$), and salt-and-pepper noise ($P = 0.001, 0.005, 0.01$). Reporting both models under the same attacks compares the two released configurations, which differ in the noise-layer curriculum and also in optimizer, learning rate, batch size, and training length, so this is a stage comparison rather than a controlled ablation of the noise layer described in Section 4.2.

Because the attacks are applied after embedding, the stego frame itself is unchanged by them. Each model's stego quality is therefore the same under every attack, at 45.68 dB PSNR and 0.9848 SSIM for the Base model and 40.42 dB and 0.9543 for the Robust variant, so Table 8 reports only

**TABLE 8: Per-attack robustness evaluation, Base model and Robust variant. Stego quality is constant across attacks: 45.68 dB/0.98 for Base, 40.42 dB/0.95 for Robust**

| Attack | Parameter | Robust Rec. PSNR (dB) | Robust Rec. SSIM | Robust BER (%) | Base Rec. PSNR (dB) | Base SSIM | Base BER (%) |
|---|---|---|---|---|---|---|---|
| Clean | — | 33.48 | 0.9648 | 1.66 | 39.25 | 0.9665 | 0.78 |
| JPEG | Q=95 | 27.39 | 0.8545 | 4.11 | 21.91 | 0.6569 | 6.45 |
| JPEG | Q=85 | 18.98 | 0.5462 | 9.47 | 13.82 | 0.4076 | 19.92 |
| JPEG | Q=70 | 13.97 | 0.4309 | 20.46 | 12.25 | 0.3888 | 26.65 |
| Gaussian noise | σ=0.005 | 31.53 | 0.9258 | 2.09 | 29.10 | 0.8049 | 2.98 |
| Gaussian noise | σ=0.01 | 27.85 | 0.7747 | 3.45 | 21.79 | 0.4663 | 7.03 |
| Gaussian noise | σ=0.02 | 21.52 | 0.4297 | 7.52 | 14.27 | 0.1835 | 17.39 |
| Gaussian blur | σ=0.3 | 31.85 | 0.9545 | 2.22 | 37.04 | 0.9567 | 1.06 |
| Gaussian blur | σ=0.5 | 16.28 | 0.6461 | 23.06 | 16.28 | 0.6419 | 22.58 |
| Gaussian blur | σ=1.0 | 12.25 | 0.4333 | 26.68 | 12.24 | 0.4317 | 26.68 |
| Salt-and-pepper | *P*=0.001 | 31.73 | 0.9388 | 1.74 | 17.91 | 0.7522 | 4.86 |
| Salt-and-pepper | *P*=0.005 | 26.97 | 0.8220 | 2.62 | 11.20 | 0.3219 | 17.56 |
| Salt-and-pepper | *P*=0.01 | 23.52 | 0.6812 | 3.99 | 8.58 | 0.1518 | 28.34 |

BER: Bit error rate

the recovery metrics, which are what the attacks affect. The Robust variant's stego quality is 5.26 dB below the Base model's, and its clean-condition recovery falls from 39.25 dB to 33.48 dB, with BER rising from 0.78% to 1.66%. This is the cost of the robustness curriculum, and it is reported here so that the trade-off between the two variants is explicit.

The Robust variant recovers better under every distortion family except mild blur. The largest gains are under salt-and-pepper noise, where the Base model falls to 17.91 dB at $P = 0.001$ and 8.58 dB at $P = 0.01$, while the Robust variant holds 31.73 dB and 23.52 dB, and under Gaussian noise at $\sigma = 0.02$, where recovery improves from 14.27 dB to 21.52 dB. Under JPEG, the gain is smaller but consistent: 5.48 dB at Q = 95 and 5.16 dB at Q = 85. The exception is Gaussian blur at $\sigma = 0.3$, where the Base model reaches 37.04 dB against 31.85 dB, and at the two stronger blur settings the two models are indistinguishable. The curriculum therefore buys tolerance to additive noise and compression at a cost of 5.26 dB in stego quality and 5.77 dB in clean recovery, without improving resistance to blur.

Within the Robust variant alone, the mildest noise, blur, and salt-and-pepper settings all sit within about 2 dB of the clean recovery figure, while JPEG at Q = 95 is already 6.09 dB lower, and performance falls steadily as the attacks strengthen. Blur has the sharpest threshold of the four attack types, with recovery falling from 31.85 dB at $\sigma = 0.3$ to 16.28 dB at $\sigma = 0.5$, more than 15 dB for a small parameter change.

JPEG remains a clear weakness, dropping to 27.39 dB at Q = 95 and 18.98 dB at Q = 85, a known limitation of spatial-domain encoder-decoders that embed in high-frequency components removed by JPEG quantization [20]. JPEG robustness is therefore not claimed. In practical terms, the method is suited to lossless or lightly processed transmission channels; it would not survive upload to a platform that re-encodes video with lossy compression. The JPEG results in Table 8 were obtained using the same differentiable JPEG approximation applied in the training noise layer, not a standard JPEG encoder. Evaluation with a standard encoder is left to future work.

All four attack families evaluated here were also used in the training noise layer of Section 3.4. However, the noise curriculum reached a strength of 0.51 at the selected checkpoint, not full strength, so the model saw a narrower range of settings than the configuration allows: JPEG Q from 72 to 95, Gaussian noise $\sigma$ from 0.001 to 0.041, Gaussian blur $\sigma$ from 0.3 to 0.91, and salt-and-pepper p from 0.01 to 0.056. Four of the twelve test settings fall outside this range. JPEG at Q = 70 is just below the lowest quality seen in training, and Gaussian blur at $\sigma = 1.0$ is just above the strongest blur seen, so both are harder than anything the model was trained on, although by a small margin. Salt-and-pepper at $P = 0.001$ and $P = 0.005$ are weaker than the lowest level used in training. Table 8 therefore covers both seen and unseen attack strengths, with the two harder settings lying only slightly beyond the training range. The model was not tested against attack types missing from training, so its behavior under such attacks is not known.

Among the 24 BPP video works reviewed, En-VStegNET [13] reports clean-condition quality only, with no attack evaluation, and Weng *et al.* [14] report a single high-quality MP4 setting in PSNR alone. DRANet [30] reports

only MSE training loss under three distortions without PSNR, SSIM, or BER. In the image domain, robustness at this payload has been studied more widely, for example Ying *et al.* [31] reporting 26.69 dB under JPEG QF = 70, and several INN-based and generative methods [32]–[37], such as RIHINNet [37] reaching 30.85 dB and 0.9124 SSIM under random JPEG and DERIS [35] reaching 30.24 dB under Gaussian noise, though all operate in the image domain under different paradigms. To the best of the authors' knowledge, no prior 24 BPP video steganography work reports a systematic multi-attack robustness characterization in PSNR, SSIM, and BER, so Table 8 is offered as a reference point for this payload.

## 5.6. Ablation Studies

An ablation study measured the contribution of the two main architectural blocks, the RDB and CBAM.

### 5.6.1. Architectural components: RDB and CBAM

This ablation measures the contribution of the two main blocks: The RDB, which provides depth and feature reuse, and CBAM, which re-weights features along the channel and spatial axes. Two reduced models were built: One without RDB, in which each RDB is replaced by a plain residual convolution block, and one without CBAM, in which all attention modules are removed. Both were trained on the same data with the same batch size, loss weights, and random seed as the full model, and each was selected with the same checkpoint rule as in Section 4.2. The three models were then evaluated on the Vimeo-90K test set under clean conditions (Table 9).

Removing RDB caused the largest degradation. Recovery PSNR fell by 3.36 dB and BER rose from 0.78% to 1.47%, while stego PSNR fell by only 0.36 dB. Removing CBAM reduced stego PSNR by 0.85 dB and recovery PSNR by 0.70 dB, while BER remained almost unchanged (0.83% against 0.78%). These differences are consistent across the test set: The full model achieves a higher recovery PSNR than the model without RDB in 98.4% of the test frames and higher than the model without CBAM in 80.9%, and both differences are statistically significant (paired Wilcoxon signed-rank test, $P < 0.001$). Both blocks therefore contribute to the final model, with RDB having the larger effect.

### 5.6.2. Noise layer

The noise layer was evaluated by comparing the Base model with the Robust variant rather than by removal. The Robust variant adds noise-layer training and further training to the Base model, so the comparison is not a controlled ablation, and Table 8 shows the effect: Clean recovery falls from 39.25 dB to 33.48 dB while recovery under attack improves in nine of the twelve configurations, is lower under Gaussian blur at $\sigma = 0.3$ and is unchanged at $\sigma = 0.5$ and 1.0, as noted in Section 5.5.

Ultimately, this trade-off comes from where the secret data are stored. The payload is contained in these (mostly high-frequency) pixel details, which are the very details JPEG's DCT quantization eliminates. This is why compression degrades recovery quality so sharply, while the visual quality of the stego frame stays almost the same. Making these details survive compression is not simply a matter of training for longer. Real JPEG provides the encoder with no useful gradient, so the encoder can only be directed toward more compression-tolerant features through the differentiable JPEG approximation used in the noise layer. The same fine detail that JPEG discards did not produce a classical statistical trace that separates cover from stego (Section 5.4).

## 5.7. Temporal Consistency and Video Codec Compression

Both evaluations use the 30 video sequences of the DAVIS 2017 validation set [38]. For this test, the sequences were paired by length so that each secret sequence is hidden in full, frame by frame, in one continuous cover sequence. This gives 15 pairs. This pairing differs from the frame pairs used for the cross-dataset evaluation in Table 7, yet the uncompressed results of the two agree closely: Stego PSNR 42.93 dB and recovery PSNR 36.88 dB for the temporal and codec test, against 43.41 dB and 36.41 dB for the cross-dataset evaluation in Table 7. All results use the Base model, and frames were resized as described in Section 5.3 and stored without loss.

Temporal consistency. All measures compare each frame with the frame before it. The first follows the embedding

**TABLE 9: Effect of removing RDB or CBAM, measured on the Vimeo-90K test set under clean conditions**

| Configuration | Stego PSNR (dB) | Recovery PSNR (dB) | Stego SSIM | Recovery SSIM | BER (%) |
|---|---|---|---|---|---|
| Without RDB | 45.32 | 35.89 | 0.9828 | 0.9630 | 1.47 |
| Without CBAM | 44.83 | 38.55 | 0.9824 | 0.9677 | 0.83 |
| Full model | 45.68 | 39.25 | 0.9848 | 0.9665 | 0.78 |

RDB: Residual dense blocks, CBAM: Convolutional block attention module, BER: Bit error rate

difference, defined as the stego frame minus the cover frame ($D_t = S_t - C_t$). Its temporal mean absolute error on the 0–255 scale, tMAE = mean $|D_t - D_{(t-1)}|$, shows how much the embedding pattern changes from one frame to the next; a small value means little flicker. The second compares the frame-to-frame PSNR and SSIM of the stego sequence ($S_t$ against $S_{(t-1)}$) with the same values for the cover sequence ($C_t$ against $C_{(t-1)}$). These values are naturally low for moving content, and embedding that added flicker would make the stego values differ from the cover values. Because these measures compare pixels at the same positions and do not follow motion, two standard motion-based measures were also added. The warping error $E_{warp}$ [39] moves the previous frame along the motion of the cover (its optical flow) so that it lines up with the present frame, and measures the error that remains, ignoring pixels that appear or disappear between frames. It is computed for both the cover and stego sequences using the same cover motion, so any increase for the stego sequence comes from the embedding. tOF [40] measures how much the motion of the stego sequence differs from that of the cover, in pixels. For both measures, lower values mean better temporal consistency. Optical flow was estimated with the DIS method in OpenCV.

As shown in Table 10, the embedding difference changes by only 1.08 intensity levels on average between consecutive frames (0.42% of the 0–255 range). The frame-to-frame PSNR of the stego sequences matches that of the covers (20.81 against 20.80 dB, within 0.06 dB in every pair), and the frame-to-frame SSIM differs by only 0.004 on average (at most 0.018 in any pair). With motion taken into account, the warping error of the stego sequences is only 2.9% above that of the covers (3.162 against $3.073 \times 10^{-3}$), and the motion in the stego sequences differs from that in the covers by only 0.172 pixels on average (tOF). Embedding therefore leaves the frame-to-frame behavior of the cover video essentially unchanged.

Video codec compression. Because the method is suited to lossless transmission channels (Section 5.5), each complete stego sequence was encoded as an H.264/AVC video and as an H.265/HEVC video using the lossless mode of each codec, then decoded and passed to the revealing network. For both codecs and in every pair, the decoded stego frames were identical to the stego frames before encoding, and recovery was the same as without coding: 36.88 dB PSNR, 0.9788 SSIM, and 1.69% BER for the recovered secret, with 42.93 dB PSNR for the stego frames. The stego video can therefore be stored and delivered as an H.264 or H.265 file encoded in lossless mode without any effect on the recovered secret.

**TABLE 10: Temporal consistency of the stego sequences (DAVIS 2017, 15 pairs, mean over pairs)**

| Measure | Cover | Stego |
|---|---|---|
| Frame-to-frame PSNR (dB) | 20.80 | 20.81 |
| Frame-to-frame SSIM | 0.5576 | 0.5533 |
| tMAE of embedding difference (intensity levels) | – | 1.08 |
| Warping error $E_{warp}$ ($\times 10^{-3}$) | 3.073 | 3.162 |
| tOF (pixels) | – | 0.172 |

## 6. CONCLUSION

In this paper, an encoder-decoder model for high-capacity video steganography is proposed to embed a full RGB secret frame into a cover frame, frame by frame, at 24 BPP. The encoder receives a 6-channel input, containing the cover frame and secret frame, and learns a small additive residual (stego = cover + tanh(residual) × 0.1), with RDBs and CBAM attention to preserve fine detail and keep the quality of the stego and recovered frames high. On the Vimeo-90K test set, the method achieved a stego PSNR of 45.68 dB, a recovery PSNR of 39.25 dB and a BER of 0.78%. It also kept its performance on unseen content, with a stego PSNR of 44.27 dB and a recovery PSNR of 37.51 dB on UCF-101, and 43.41 dB and 36.41 dB on DAVIS 2017, and it embeds at 24.8 FPS on a single T4 GPU.

In addition to visual quality, the stego frames successfully passed 6 out of 7 classical statistical tests and were only partially detected by StegExpose with a 23.0% true positive rate and 0% false-positive rate (a small but non-zero statistical trace at this high payload). A robustness analysis was carried out on this 24 BPP video-in-video payload using the Robust variant, whose stego quality is 40.42 dB, with good resistance to mild noise and blur, and a clear weakness against JPEG compression.

The method embeds and recovers each frame on its own, which keeps recovery independent of frame order and robust to frame loss. The temporal evaluation on DAVIS 2017 showed that embedding leaves the frame-to-frame behavior of the cover video essentially unchanged, with a warping error only 2.9% above that of the covers and a motion difference of 0.172 pixels on average, and that the stego video can be stored as a lossless H.264 or H.265 file without any effect on the recovered secret.

The main direction for future research is compression robustness. The secret is carried as fine spatial detail, and

JPEG's DCT quantization discards exactly that detail; since the quantization step is not differentiable, the encoder cannot be trained against real compression directly. The noise layer used here relies on a differentiable JPEG approximation, mixed with clean batches. Adding a real-JPEG branch to that mini-batch mix, as in MBRS [20], is the first step. Since lossless H.264 and H.265 coding already preserves the secret, the next step is to extend the noise layer to the lossy settings of these codecs. A second direction is temporal modeling. The encoder could use neighboring frames when computing the residual, so that the residual changes smoothly from one frame to the next. The drawback is that frames would then depend on each other, and the present design would lose its ability to recover each frame on its own. A third direction is evaluating and improving resistance to modern learned steganalyzers. These detectors learn their own features from data, unlike the classical tools used here, which test fixed statistical properties, and the present design makes no claim of resistance to them at this payload. In conclusion, the results show that residual dense learning with attention is a promising approach for hiding a full frame inside another at high quality, and provide a reference point for robustness at this payload in video-in-video hiding.

## REFERENCES


[1] K. M. Hosny, M. A. Zaki, N. A. Lashin, M. M. Fouda and H. M. Hamza. "Multimedia security using encryption: A survey". *IEEE Access*, vol. 11, pp. 63027-63056, 2023.

[2] V. Reshma, S. Joseph Gladwin and C. Thiruvenkatesan. Pairing-free CP-ABE based cryptography combined with steganography for multimedia applications. In: "*2019 International Conference on Communication and Signal Processing (ICCSP)"*. IEEE, Chennai, pp. 501-505, 2019.

[3] A. K. Singh, X. Liu, H. Wang and H. Ko. "Recent advances in multimedia security and information hiding". *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 2, p. e4193, 2021.

[4] N. F. Sahib, S. H. Hashem and E. F. Naser. "Review of video steganography by using deep learning methods: Datasets, techniques, and evaluations". *Journal of Soft Computing and Computer Applications*, vol. 2, no. 2, p. 1022, 2025.

[5] Cisco Systems Inc., "*Cisco Annual Internet Report (2018-2023) White Paper 2020*." Available from: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html [Last accessed on 2026 Aug 15].

[6] Sandvine Inc., "*Like our Cosmos, the Internet is a Vast, Expanding Universe Peering into its Mysteries Reveals its Current and Future Impact on Networks and App QoE 2024*". Available from: https://www.sandvine.com/hubfs/Sandvine_Redesign_2019/Downloads/2024/GIPR/GIPR%202024.pdf [Last accessed on 2026 Mar 07].

[7] C. Mou, Y. Xu, J. Song, C. Zhao, B. Ghanem and J. Zhang. "Large-Capacity and Flexible Video Steganography via Invertible Neural Network. In: "*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)"*. IEEE, Vancouver, BC, Canada, pp. 22606-22615, 2023.

[8] S. Xu, Z. Li, Z. Zhang and J. Liu. "An end-to-end robust video steganography model based on a multi-scale neural network". *Electronics (Switzerland)*, vol. 11, no. 24, p. 4102, 2022.

[9] K. Hu, Z. Hu, Q. Zhu, X. Wang and X. Wang. StegaVideo: Robust High-Resolution Video Steganography with Temporal and Edge Guidance. In: "*Pacific Graphics Conference Papers and Posters"*. The Eurographics Association, Switzerland, 2024.

[10] J. Zhu, R. Kaplan, J. Johnson and L. Fei-Fei. HiDDeN: Hiding Data With Deep Networks. In: "*Computer Vision - ECCV 2018"*. Springer, Cham, pp. 682-697, 2018.

[11] H. Chai, Z. Li, F. Li and Z. Zhang. "An end-to-end video steganography network based on a coding unit mask". *Electronics (Switzerland)*, vol. 11, no. 7, p. 1142, 2022.

[12] P. Malathi and T. Gireesh Kumar. "Deep steganographic approach for reliable data hiding using convolutional neural networks and adaptive loss optimization". *Scientific Reports*, vol. 15, no. 1, p. 45692, 2025.

[13] A. Jaiswal, S. Kumar and A. Nigam. En-VStegNET: Video Steganography using Spatio-Temporal Feature Enhancement with 3D-CNN and Hourglass. In: "*Proceedings of the International Joint Conference on Neural Networks (IJCNN)"*. IEEE, Glasgow, UK, pp. 1-8, 2020.

[14] X. Weng, Y. Li, L. Chi and Y. Mu. High-Capacity Convolutional Video Steganography with Temporal Residual Modeling. In: "*ICMR 2019 - Proceedings of the 2019 ACM International Conference on Multimedia Retrieval"*. Association for Computing Machinery Inc., New York City, pp. 87-95, 2019.

[15] J. Jing, X. Deng, M. Xu, J. Wang and Z. Guan. HiNet: Deep Image Hiding by Invertible Network. In: "*Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*". IEEE, Montreal, QC, Canada, pp. 4733-4742, 2021.

[16] Y. Zhang, Y. Tian, Y. Kong, B. Zhong and Y. Fu. Residual Dense Network for Image Super-Resolution. In: "*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)"*. IEEE, Salt Lake City, UT, USA, pp. 2472-2481, 2018.

[17] S. Woo, J. Park, J. Y. Lee and I. S. Kweon. CBAM: Convolutional Block Attention Module. In: "*Computer Vision - ECCV 2018"*. Springer, Cham, pp. 3-19, 2018.

[18] T. Xue, B. Chen, J. Wu, D. Wei and W. T. Freeman. "Video enhancement with task-oriented flow". *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106-1125, 2019.

[19] R. Zhang, P. Isola, A. A. Efros, E. Shechtman and O. Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In: "*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)"*. IEEE, Salt Lake City, UT, USA, pp. 586-595, 2018.

[20] Z. Jia, H. Fang and W. Zhang. MBRS: Enhancing Robustness of DNN-Based Watermarking by Mini-Batch of Real and Simulated JPEG Compression. In: "*MM 2021 - Proceedings of the 29th ACM International Conference on Multimedia"*. Association for Computing Machinery Inc., New York City, pp. 41-49, 2021.

[21] B. Boehm. "*StegExpose - A Tool for Detecting LSB Steganography, 2014*". Available from: https://arxiv.org/abs/1410.6656 [Last accessed on 2026 Oct 16].

[22] M. Dalal and M. Juneja. "A secure video steganography scheme using DWT based on object tracking". *Information Security Journal*, vol. 31, no. 2, pp. 196-213, 2022.

[23] F. Ernawan. “An improved hiding information by modifying selected DWT coefficients in video steganography”. *Multimedia Tools and Applications*, vol. 83, no. 12, pp. 34629-34645, 2024.

[24] C. Zhang, P. Benz, A. Karjauv and G. Sun. UDH: Universal Deep Hiding for Steganography, Watermarking, and Light Field Messaging. In: “*Advances in Neural Information Processing Systems”*. Curran Associates, United States, pp. 10223-10234, 2020.

[25] S. Baluja. Hiding Images in Plain Sight: Deep Steganography. In: “*Advances in Neural Information Processing Systems 30*”. NIPS, Long Beach, CA, USA, pp. 2069-2079, 2017.

[26] A. Mishra, S. Kumar, A. Nigam and S. Islam. VStegNET: Video Steganography Network using Spatio-Temporal features and Micro-Bottleneck. In: “*Proceedings of the British Machine Vision Conference (BMVC)”*. BMVA Press, Cardiff, UK, 2019.

[27] G. Li, S. Li, Z. Luo, Z. Qian and X. Zhang. Purified and Unified Steganographic Network. In: “*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)”.* Seattle, WA, USA, pp. 27569-27578, 2024.

[28] H. Liu, T. Xiang, S. Guo, H. Li, T. Zhang and X. Liao. “Erase and repair: An efficient box-free removal attack on high-capacity deep hiding”. *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 5229-5242, 2023.

[29] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli. “Image quality assessment: From error visibility to structural similarity”. *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, 2004.

[30] J. Cui, L. Zheng, Y. Yu, Y. Lin, H. Ni, X. Xu and Z. Zhang. “Deeply-recursive attention network for video steganography”. *CAAI Transactions on Intelligence Technology,* vol. 8, no. 4, pp. 1507-1523, 2023.

[31] Q. Ying, H. Zhou, X. Zeng, H. Xu, Z. Qian and X. Zhang. Hiding Images into Images with Real-World Robustness. In: “*Proceedings of the IEEE International Conference on Image Processing (ICIP)”.* IEEE, Bordeaux, France, pp. 111-115, 2022.

[32] Y. Xu, C. Mou, Y. Hu, J. Xie and J. Zhang. Robust Invertible Image Steganography. In: “*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)”*. IEEE, New Orleans, LA, USA, pp. 7875-7884, 2022.

[33] H. Yang, Y. Xu, X. Liu and X. Ma. “PRIS: Practical robust invertible network for image steganography”. *Engineering Applications of Artificial Intelligence*, vol. 133, Part D, p. 108419, 2024.

[34] L. Chen, B. Feng, Z. Xia, W. Lu and J. Weng. “Robust generative steganography for image hiding using concatenated mappings”. *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 5952-5965, 2025.

[35] D. Cheng, Y. Kong, Y. Mao and L. Bian. “Denoising and enhancement for robust image steganography”. *PeerJ Computer Science,* vol. 11, p. e3368, 2025.

[36] L. Huang, L. Huo, Z. Gan and X. He. “*FIIH: Fully Invertible Image Hiding for Secure and Robust*, 2024”. Available from: https://arxiv.org/abs/2407.17155 [Last accessed on 2026 Oct 16].

[37] X. Jin, C. Pan, Z. Cheng, Y. Dong and Q. Jiang. “RIHINNet: A robust image hiding method against JPEG compression based on invertible neural network”. *IET Image Processing*, vol. 18, no. 13, pp. 3801-3814, 2024.

[38] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung and L. Van Gool. “*The 2017 DAVIS Challenge on Video Object Segmentation”. arXiv* [preprint]*, arXiv:*1704.00675, 2017.

[39] W. S. Lai, J. B. Huang, O. Wang, E. Shechtman, E. Yumer and M. H. Yang. Learning blind video temporal consistency. In: V. Ferrari, M. Hebert, C. Sminchisescu and Y. Weiss, (Eds.). “*Computer Vision - ECCV 2018*, *Lecture Notes in Computer Science”*. vol. 11219. Springer, Cham, pp. 179-195, 2018.

[40] M. Chu, Y. Xie, J. Mayer, L. Leal-Taixé and N. Thuerey. “Learning temporal coherence via self-supervision for GAN-based video generation”. *ACM Transactions on Graphics*, vol. 39, no. 4, pp. 75:1-13, 2020.

[41] S. H. O. Hashemi, M. H. Majidi and S. Khorashadizadeh. “A new architecture based resnet for steganography in color images”. *Multimedia Tools and Applications*, vol. 84, pp. 16229-16248, 2025.

[42] G. F. Vergara, P. Giacomelli, A. L. Marques Serrano, F. L. L. De Mendonça, G. A. P. Rodrigues, G. D. Bispo, V. P. Gonçalves, R. D. O. Albuquerque and R. T. De Sousa Júnior. “Stego-STFAN: A novel neural network for video steganography”. *Computers*, vol. 13, no. 7, p. 180, 2024.

[43] S. P. Lu, R. Wang, T. Zhong and P. L. Rosin. Large-Capacity Image Steganography Based on Invertible Neural Networks. In: “*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)”*. IEEE, Nashville, TN, USA, pp. 10816-10825, 2021.