StegaNeRV: Video Steganography using Implicit Neural Representation

Monsij Biswal*, Tong Shao†, Kenneth Rose*, Peng Yin†, Sean McCarthy†

*Department of Electrical and Computer Engineering
University of California, Santa Barbara
Santa Barbara, CA 93106, USA
{mbiswal, kenrose}@ucsb.edu

†Dolby Laboratories, Inc.
432 Lakeside Dr
Sunnyvale, CA 94085, USA
{tong.shao, pyin, sean.mccarthy}@dolby.com

Abstract

Numerous studies have recently advanced the state-of-the art for representing videos through an implicit neural network (INR). As these models become increasingly ubiquitous, there is a growing demand for concealing data within INR reconstructed videos such as for storing content metadata and sensitive licensing information. In this paper, we explore a new space in video steganography, hiding a distinct image within each RGB frame output by an INR. We propose a joint training strategy of a U-Net based steganographic decoder with an INR model for video. Experimental results show that hidden images can be embedded and subsequently reconstructed with high fidelity while preserving the quality of the cover frames. Furthermore we demonstrate that by introducing an attention module which emphasizes hiding within the edges and rich texture patches in the cover frame, secret images can be reconstructed with superior quality and can also be concealed at greater resolutions.

1. Introduction

Implicit neural representations (INRs) have been an emerging topic of research in recent years, owing to their potential to learn continuous signal mapping from a regular grid of coordinates to their corresponding values. For instance, each spatial pixel coordinate (x, y) in an image is associated with an RGB pixel value. Similarly for a video, each spatio-temporal coordinate (x, y, t) has its corresponding color pixel, where t indexes each frame across time. Therefore an image or a video can be formulated as a mapping from a set of coordinates to its corresponding attribute. Given the generality in its formulation, INRs have been successfully applied in a variety of applications including reconstruction of 3D scenes [39, 47], shapes [45, 58] and an abundance of 3D tasks [30, 40, 67]. Furthermore, the authors in [52] illustrated that by leveraging periodic activation functions, INRs can faithfully reconstruct signals with high-frequency information such as those in audio [21, 52], images [9, 51, 53] and videos [7, 26, 29].

In contrast to most INR-based pixel-wise image representations [52, 53] where each RGB pixel is predicted as a function of spatial pixel coordinates, the authors in [7] proposed a model performing one-shot prediction, called Neural Representation for Videos (NeRV), by implicitly learning a function $f_\theta : \mathbb{R} \rightarrow \mathbb{R}^{H\text{x}W\text{x}3}$ which maps a given normalized frame index, $t \in \mathbb{R}$ directly to the entire RGB video frame. Given the large number of pixels in high-resolution videos, this structural change paved the way for a new paradigm, introducing considerable savings - both in terms of encoding and decoding speed. Several recent works have focused on a variety of aspects for improving INR video representation including adoption of a patch-based approach [2], enhanced motion modeling [71, 72] as well as disentangling spatial and temporal pixel correlation with fewer model parameters [29]. As an outcome of these collective endeavors, the rate-distortion performance gap between such INR-based models and traditional state-of-the-art video codecs namely Versatile Video Coding (VVC) [4] and High Efficiency Video Coding (HEVC) [54] or popular learning-based methods such as DVC [32] and DCVC [28] has been steeply reducing, although far from being competitive in its current state.

Steganography is a well known procedure for hiding data unnoticeably within a cover medium. This is largely different from cryptography where the encoded information still resides in plain sight. With videos being a popular choice for sharing media content - accounting for over 50% of the overall internet traffic, video steganography has been extensively investigated - hiding images, audio and text [18, 24, 49, 60] within a cover video. Classical approaches for video steganography can be broadly classified into two categories based on their domain of application. In the spatial domain, Least Significant Bits (LSBs) of the cover video are either replaced [56] (LSB replacement) or altered [38] (LSB matching) depending on the bits of the hidden data. On the other hand, in the frequency domain, cover frames are first decomposed into a set of transform coefficients through a decorrelating transform such as Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT). The bits from the hidden information are then embedded into these coefficients, leading to an unobstrusive and subtle hiding mechanism [23, 48, 55].

During the initial explorations of utilizing deep learning for image steganography, neural networks were mostly designed to optimize hiding messages within the LSBs of the cover image [19, 20]. One of the first end-to-end framework for deep image steganography using convolutional neural network (CNN) was proposed in [3]. In this work, while the hiding network concealed the secret image within the cover image (generating a so-called *container image*), the reveal network recovered the secret image using the container image. For video steganography, although it is natural to formulate it as a collection of image steganography tasks, this approach is largely suboptimal since it completely ignores temporal correlation among the frames. Following this, a 2-dimensional CNN based model was proposed for video-in-video steganography [63], where based on the type of hidden frame (reference or sparse inter-frame residual) separate networks were trained for hiding data within the frames of the cover video. Moreover, the authors in [41] proposed a 3-dimensional CNN for video steganography, purposefully designed to account for spatial and temporal features [62].

In this paper, we begin by exploring what happens if we apply traditional steganography techniques on the cover frames followed by learning the entire *container video* employing NeRV [7], with an expectation to be able to recover the hidden information through the INR reconstructed video. Here, the *container video* is a regular video with an hidden image per frame. Formally, given a sequence of K cover frames $\{I_{c_i}\}_{i=1}^{K}$ and an equal number of images to be hidden $\{M_{g_i}\}_{i=1}^{K}$, per-frame LSB steganography is performed resulting in a sequence of stego-frames $\{I_{s_i}\}_{i=1}^{K}$. This group of frames $\{I_{s_i}\}_{i=1}^{K}$ forms the training set for the NeRV model. After training the model, we process each frame for recovering the hidden image, the results of which are shown in Fig. 1. We implement spatial domain Universal Wavelet Relative Distortion (S-UNIWARD) [17], a well known digital image steganography technique which measures embedding distortion in a fixed domain independent of the target domain. Although NeRV is able to capture and reconstruct contents of the frame precisely, tiny perturbations such as those introduced by LSB steganography are not replicated accurately enough, causing hidden recovery to fail. Therefore the recovered hidden images $\{M_{r_i}\}_{i=1}^{K}$ have no resemblance to their ground truth counterparts.

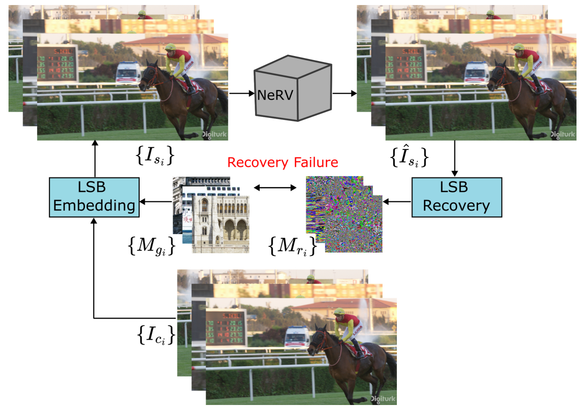


Figure 1. Failure of LSB steganography when container frames are reconstructed by NeRV.

Recently, there have been numerous efforts to combine steganography with implicit neural representations, particularly those specific to Neural Radiance Fields (NeRF) [8, 12, 27, 34]. This paper introduces an attention-based methodology to integrate video steganography within the learning framework of NeRV [7], dubbed StegaNeRV. To the best of our knowledge, this is the first work to consider the objective of video steganography where hidden information is concealed within container video frames reconstructed by an implicit neural network. The primary contributions of this paper can be summarized as follows:

- We explore the new domain of NeRV video steganography, where we hide a distinct image in each frame of the cover video.
- We propose stage-wise gradient scaling across different stages of NeRV, gradually perturbing the weights aligned with a steganographic objective.
- We utilize an attention-guided approach to emphasize concealing information within regions of rich texture in the cover frame, thereby enabling us to hide larger images.

2. Related Work

2.1. Implicit Neural Representations for Videos

Implicit neural representations have been known to be a versatile and flexible mode of representing a wide variety of signals such as images [14, 53], video [35, 71], audio [25, 57] including 3D shapes and scenes [37, 46, 58]. For videos, the authors in [7] proposed NeRV, which performs a one-shot RGB frame prediction instead of pixel-wise implicit reconstruction. Although the architecture was desirable in terms of encoding and decoding speed, a separate model was required to be trained for each video. Additionally, due to coupling of spatial and temporal contexts, a significant number of model parameters were redundant, resulting in an inflated model size. Several subsequent works addressed these critical limitations as well as established new benchmarks. One of the changes proposed in [29] demonstrated that by providing temporal context information to each NeRV block, better content reproduction can be achieved with a lower training time. Furthermore, D-NeRV [16] was proposed to learn an implicit representation for a diverse set of videos by modelling content dependent features and motion information separately. Apart from video representation, INRs have also been used for other tasks such as video denoising [13], frame interpolation [10], action recognition [16] and video generation [50, 68].

2.2. Video Steganography

The widespread use of videos combined with its inherent redundancy in both space (intra frame) and time (inter frame) create ideal conditions for large capacity data hiding. In the spatial domain, video steganography is performed by strategically altering LSBs of the cover frame pixels. For instance, [56] proposed utilizing polynomial equations for determining the locations of the pixels to be modified. Furthermore, a preprocessing stage for encrypting data prior to its embedding in the cover frame was explored in [66] which exhibited greater robustness and security. In the frequency domain, [43] proposed embedding encrypted messages within DCT coefficients of Y, U, V components of the cover frame. The original message was first encoded using hamming codes prior to data hiding which was shown to improve data security over direct approaches. [44] further demonstrated that by concealing the hamming codes selectively within DCT and DWT coefficients of the blocks corresponding to moving objects in the cover video, greater robustness, imperceptibility, and embedding capacity can be achieved. There are several works that investigated video steganography in the compressed domain. Such methods are often designed to work with a certain video codec. While [59] explored information hiding by manipulating block decisions of HEVC [54], the authors in [31] proposed embedding secret message by adding an error matrix to 4x4 quantized discrete sine transform (DST) coefficients. Drifts in intra frame prediction were prevented by restricting such modifications to a certain class of blocks only.

With the rise of deep learning, several methods were proposed that achieved data hiding and their subsequent recovery by means of deep neural networks. Weng *et al.* proposed the first framework for hiding a video within another video [63] via separate hiding and recovery networks for reference and residual frames of the secret video. Additionally, Generative Adversarial Networks (GANs) have also been employed for video steganography where the discriminator assumes the role of a steganalysis classifier, enabling the generator to hide data with greater imperceptibility [65, 70]. [69] introduced an attention mechanism alongside GANs which was shown to be robust against noise layers such as compression, cropping and scaling. Along these lines, [5] proposed using a GAN assisted by a Coding Unit (CU) mask generated by a video encoder to hide random bits within certain key frames. Invertible neural networks (INNs) have also been utilized to achieve image [33] and video [42] steganography, particularly attractive due to its bijective nature enabling the hiding and recovery networks to share a single model with shared parameters. Experiments performed in [33] indicated that INNs have the capacity to hide multiple images within a single cover image. On the other hand, authors in [42] demonstrated concealing up to 7 secret videos within a single cover video with an added mechanism to recover them through specific keys.

3. Method

In this section, we describe two approaches for achieving steganography within the framework of NeRV-based video representation. For simplicity, we define some notations in Tab. 1.

Table 1. Collection of symbols and their description.

Symbol	Description
t_i	Normalized frame index for i^{th} frame
$\{I_{g_i}\}$	Set of ground truth cover frames
$\{I_{s_i}\}$	Set of steganographic frames
$\{M_{g_i}\}$	Set of ground truth hidden images
$\{M_{r_i}\}$	Set of reconstructed hidden images
F_θ	NeRV model [7]
θ_0	Weights of pretrained NeRV
H_ψ	Steganographic decoder

3.1. U-Net Style Decoder with Gradient Scaling

Inspired by [27], we consider an U-Net architecture for the steganographic decoder as shown in Fig. 2. The overall training pipeline is shown in Fig. 3. Given a cover video and its corresponding NeRV model F_{θ_0}, we initialize F_θ with the pretrained model weights θ_0. We next describe the loss functions and the operation of gradient scaling.

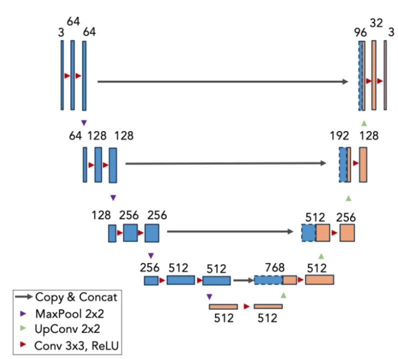


Figure 2. Model architecture for U-Net based steganographic decoder H_ψ.

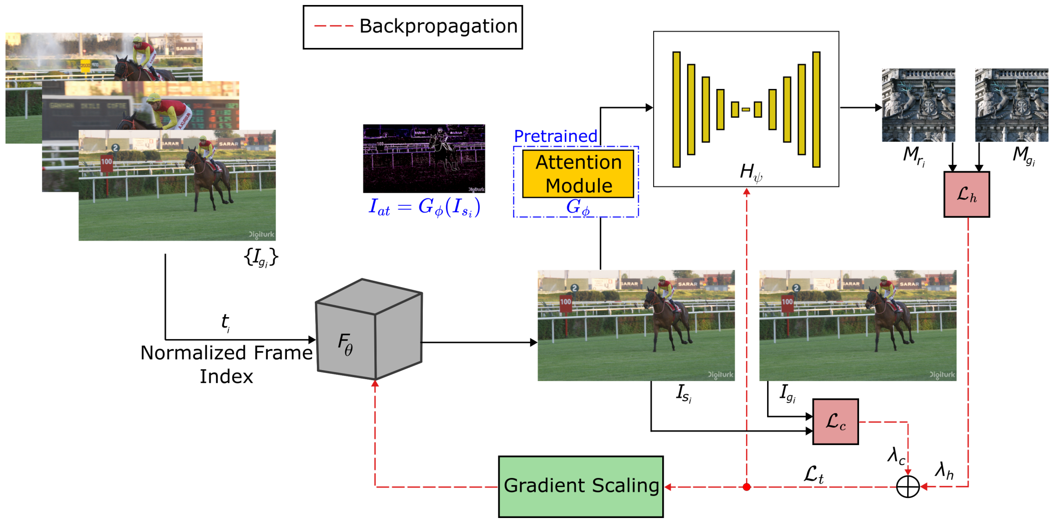


Figure 3. Overall framework for jointly training steganographic decoder with NeRV. For each normalized frame index t_i at the input, the framework produces a steganographic frame (I_{s_i}) and reveals the corresponding hidden image (M_{r_i}). For our approach without attention, I_{s_i} is directly input to the steganographic decoder (H_ψ). With attention, instead of I_{s_i}, the output of the pretrained attention module $I_{at} = G_\phi(I_{s_i})$ is input to H_ψ (highlighted in blue), which emphasizes hiding within rich texture regions and edges.

Loss Objective. Given a pair of cover and steganographic frame (I_g , I_s) with their associated ground-truth and recovered hidden images (M_g , M_r) , we compute two types of losses namely:

1. Cover frame reconstruction loss ($\mathcal{L}_c$): indicates the dissimilarity between the cover frame and steganographic frame.

$$\mathcal{L}_c = \lambda_1 \left[\frac{1}{N_c} \sum \|I_g - I_s\|_1 \right] + \mathcal{L}_{ssim,\lambda_1}(I_g, I_s) \quad (1)$$

2. Hidden image recovery loss ($\mathcal{L}_h$): indicates the dissimilarity between the ground truth hidden image and recovered hidden image.

$$\mathcal{L}_h = \lambda_2 \left[\frac{1}{N_h} \sum \|M_g - M_r\|_1 \right] + \mathcal{L}_{ssim,\lambda_2}(M_g, M_r) \quad (2)$$

where $\mathcal{L}_{ssim,\lambda_i}(x, y) = (1 - \lambda_i)[1 - SSIM(x, y)]$, with SSIM denoting the Structural Similarity Index (SSIM) evaluated between x and y. The L1 loss is obtained by averaging the absolute errors over all pixel locations: N_c & N_h being total number of pixels in the cover frame and hidden image respectively. In order to ensure the steganographic frames maintain a high degree of visual resemblance to their cover frames along with accurate recovery for hidden images, we define the overall loss ($\mathcal{L}_t$) as given in Eq. (3).

$$\mathcal{L}_t = \lambda_c \mathcal{L}_c + \lambda_h \mathcal{L}_h \quad (3)$$

The hyperparameters λ_c and λ_h balance the relative importance of cover frame reconstruction loss and hidden image recovery loss respectively. While the weights for the decoder H_ψ are updated directly using the accumulated gradients, in the case of NeRV, the gradients are adjusted as we describe next.

Gradient Scaling. It was observed that direct backpropagation of the overall loss gradient $\frac{\partial \mathcal{L}_t}{\partial \theta}$ was not working for fine-tuning NeRV for steganography. This can be explained intuitively as larger weights already contain a large amount of information for video representations and have potentially greater effect on the output quality. Therefore, the gradients for such weights should be masked out, *i.e.*, when embedding information of the hidden image through gradients, the smaller weights should be the priority to be updated as they may have additional capacity to contain more information. Furthermore, while the later stages of NeRV are based on CNNs, the initial stages consist of fully-connected layers. By virtue of their structure, each neuron in a fully-connected layer has its own weight vector whereas for a given convolutional layer, neurons share the same weights via kernels. Given these differences, we propose to perform gradient scaling for each stage separately.

For a given stage i, consider the set of weights comprising of n learnable weight parameters denoted by $\mathbf{w} = [w_1, w_2, ..., w_n]$. As an example, for a fully-connected layer of NeRV, n represents the number of neuron parameters. Using Eq. (4), we compute each element of the per-stage gradient mask vector $\mathbf{c}_i$ as:

$$c_{ij} = \frac{|w_j|^{-\alpha}}{\sum_{k=1}^{n} |w_k|^{-\alpha}} \;, 1 \leq j \leq n \quad (4)$$

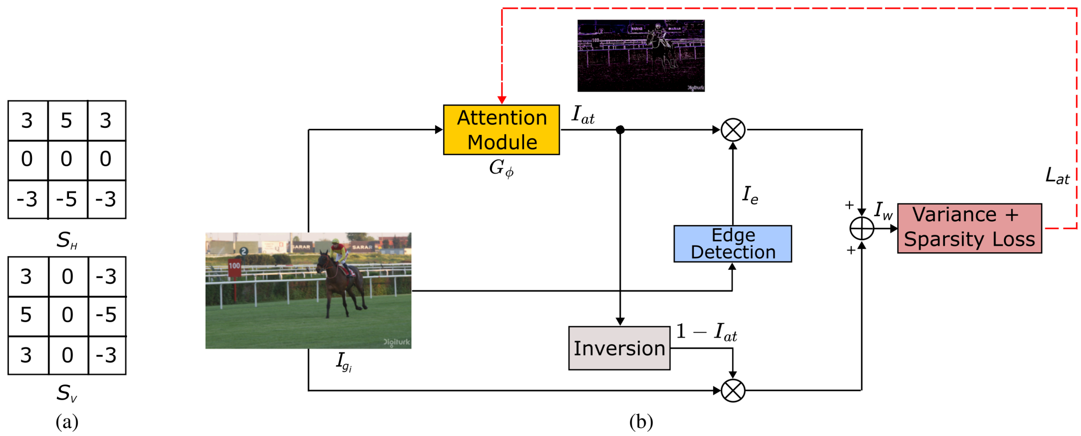


Figure 4. (a) Filter kernels for horizontal (S_H) and vertical (S_V) edge detection. (b) Training procedure for the attention module G_ϕ.

The hyper-parameter α determines the extent of disparity among the masking coefficients in a given stage. Hence the gradient mask reduces the effective gradient by scaling each component by a factor that decreases exponentially with the absolute magnitude of the weight. $\mathbf{c}$ is the gradient mask, where c_{ij} is the mask value for j^{th} parameter in i^{th} stage. In each training iteration, the gradients $\frac{\partial \mathcal{L}_t}{\partial \theta}$ are scaled as $\frac{\partial \mathcal{L}_t}{\partial \theta} \odot \mathbf{c}$ where $\odot$ denotes element-wise multiplication. Scaled gradients ensure that we slowly perturb the weights of F_θ in accordance with our dual objective of both cover content preservation and high quality message reconstruction. It is worth mentioning that we maintain a consistent gradient mask across training iterations, considering we aim to have the final steganographic video indistinguishable from the cover video.

In summary, for each cover frame I_{g_i}, we conceal a distinct image M_{g_i} resulting in a set of steganographic frames $\{I_{s_i}\}$. The decoder H_ψ operates on each I_{s_i} reconstructing the corresponding hidden image M_{r_i} as given in Eq. (5). We outline the overall training steps in Algorithm 1.

$$H_\psi(I_a) = I_b \quad ; I_a \in \{I_{s_i}\}, I_b \in \{M_{r_i}\} \tag{5}$$

3.2. Enhanced Steganography with Attention

Past works [11, 15] have shown that due to one of the weaknesses of the human visual system (HVS), it is easier to introduce unnoticeable changes within texture-rich regions and edges in an image as compared to flat and homogenous patches. In this method, we exploit this behaviour and utilize Sobel-like filter kernels for detecting edges to learn an attention module [64] to find such regions in the cover frames which are beneficial for data hiding. The structures of the filter kernels are shown in Fig. 4a. As given in Eq. (6), using these kernels, we compute the mean of the vertical and horizontal edge-maps for each input channel, thereby preserving the input channel dimensionality at the output.

Algorithm 1 U-Net style decoder with gradient scaling

Data: θ_0, $\{t_i\}$, $\{I_{g_i}\}$, $\{M_{g_i}\}$, learning rates = $[\eta_F, \eta_H]$
Initialize $F_\theta : \theta \leftarrow \theta_0$
Compute gradient mask $\mathbf{c}$
for each training iteration **do**
  for each cover frame index i **do**
    Obtain $I_{s_i} = F_\theta(t_i)$
    Reconstruct hidden image $M_{r_i} = H_\psi(I_{s_i})$
    Accumulate losses $\mathcal{L}_c$ and $\mathcal{L}_h$
  end for
  Compute overall loss $\mathcal{L}_t$
  Update F_θ with $\eta_F \cdot (\frac{\partial \mathcal{L}_t}{\partial \theta} \odot \mathbf{c})$ and H_ψ with $\eta_H \cdot \frac{\partial \mathcal{L}_t}{\partial \psi}$
end for
Output: Trained models F_θ, H_ψ

Here $*$ denotes the 2D convolution operator.

$$S(I_{g_i}) = \frac{1}{2}\Big[(I_{g_i} * S_H) + (I_{g_i} * S_V)\Big] = I_e \tag{6}$$

The architecture of the attention module [64] consists of 4 layers of convolutional neural networks with a maximum channel depth of 64. All layers utilize exponential linear unit for activation except for the last which uses sigmoid. The output attention map (I_{at}) has the same dimensions as the cover frames. As given in Eq. (7), the loss function for training the attention module, $\mathcal{L}_{at}$ has two components. While the first term guides the attention map to adjust its values in the regions of high texture *i.e.* greater pixel variance, the second term encourages a sparse representation by assigning a penalty for every non-zero element in the output attention map. We adapted the functions from [64] with a few modifications to best fit our problem.

$$\mathcal{L}_{at} = \mathbb{E}[\text{VarPool2D}_{7\,\text{x}\,7}(I_w)] + \mathbb{E}[I_{at}]^{3-2\mathbb{E}[I_{at}]} \tag{7}$$

Here $\mathbb{E}$ denotes the expectation operator and I_w is computed as a weighted combination of I_e and I_{g_i}, where I_{at}

Figure 5. Subjective results for two cover videos without using attention. The two columns on the left represent the cover (I_g) and steganographic (I_s) frames with dimensions 1920 x 1080. M_g and M_r denote the ground truth and recovered hidden images respectively of size 128 x 128.

determines the relative importance of each. The variance is computed over a sliding 2D window of size 7 x 7, thereby accounting for local intensity variations only. In Fig. 4b & Algorithm 2, we describe the training steps for the attention module. Post training, we freeze its weights and deploy it at the input of the steganographic decoder H_ψ. Following this, we jointly train F_θ & H_ψ as per Algorithm 1. Therefore, as highlighted in Fig. 3, in this approach, the output attention map of the pretrained attention module is fed to the decoder H_ψ. This operation is summarized in Eq. (8).

$$H_\psi[G_\phi(I_a)] = I_b \quad ; I_a \in \{I_{s_i}\}, I_b \in \{M_{r_i}\} \tag{8}$$

Algorithm 2 Training attention module G_ϕ

Data: Ground-truth cover frames $\{I_{g_i}\}$, learning rate η
for each training iteration **do**
  for each cover frame index i **do**
    Compute $I_{at} = G_\phi(I_{g_i})$ & $I_e = S(I_{g_i})$
    Compute $I_w = I_{at} \cdot I_e + (1 - I_{at}) \cdot I_{g_i}$
    Accumulate loss $\mathcal{L}_{at}$
  end for
  Update G_ϕ using $\eta \cdot \frac{\partial L_{at}}{\partial \phi}$
end for
Output: Trained G_ϕ

4. Experiments

4.1. Dataset

For training, we obtain 5 different 8-bit 1080p videos from the UVG dataset [36] namely *Beauty, Bosphorus, Honeybee, Jockey* and *ShakeNDry* comprising of both static and dynamic content. For each cover video, we jointly train the steganographic decoder and fine-tune the pretrained NeRV for 250 frames. Since we hide a distinct image in each cover frame, an equal number of hidden images are sampled from DIV2K dataset [1, 61] after which we randomly crop a square patch of size S where $S \in [128, 320, 512, 1088]$.

4.2. Implementation Details

We use the publicly available implementation of NeRV-L [6] for our experiments retaining all model parameters for 1080p videos: 5 NeRV blocks with up-scale factors of

Table 2. Average quantitative metrics for proposed StegaNeRV without (middle row for each cover) and with attention. Hidden images have a fixed size of 128 x 128. For each cover, the first row enlists the performance of the pretrained NeRV (F_{θ_0}). Video PSNR and SSIM compare the reconstruction quality between ground truth cover frames $\{I_{g_i}\}$ with steganographic frames $\{I_{s_i}\}$ and hidden PSNR and SSIM between ground truth hidden images $\{M_{g_i}\}$ and recovered hidden images $\{M_{r_i}\}$.

Cover	Method	Video PSNR	Video SSIM	Hidden PSNR	Hidden SSIM
	NeRV [7]	34.164	0.915	–	–
Beauty	StegaNeRV	34.166	0.915	34.240	0.968
	StegaNeRV (with attention)	34.166	0.915	35.571	0.975
	NeRV [7]	35.516	0.961	–	–
Bosphorus	StegaNeRV	35.671	0.963	34.561	0.963
	StegaNeRV (with attention)	35.586	0.962	34.710	0.971
	NeRV [7]	39.718	0.986	–	–
Honeybee	StegaNeRV	39.392	0.985	35.286	0.967
	StegaNeRV (with attention)	39.718	0.986	36.380	0.972
	NeRV [7]	35.655	0.965	–	–
Jockey	StegaNeRV	35.620	0.965	33.102	0.956
	StegaNeRV (with attention)	35.696	0.965	34.963	0.973
	NeRV [7]	35.835	0.968	–	–
ShakeNDry	StegaNeRV	35.734	0.968	31.447	0.954
	StegaNeRV (with attention)	35.875	0.968	33.792	0.964

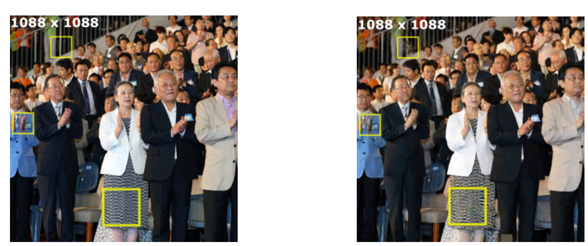
5,3,2,2,2 with $b = 1.25, l = 80$ for input embedding [7]. We set $\eta_F = 1e^{-2}$, $\eta_H = 1e^{-4}$ (varies slightly for different cover videos) and use Adam optimizer [22] with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ for both NeRV and the U-Net. For computing the gradient mask, we set $\alpha = 3$. The loss functions $\mathcal{L}_c$ and $\mathcal{L}_h$ are computed with $\lambda_1 = \lambda_2 = 0.7$. For the hyperparameters in Eq. (3), we used $\lambda_c = \lambda_h = 0.5$ for all experiments. In order to prevent detection of hidden information from a similar looking non-steganographic frame, we use a training batch size of 2 where each batch consists of the steganographic frame I_{s_i} stacked with its corresponding ground truth frame I_{g_i}.

For training G_ϕ, we use Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ and set the learning rate $\eta = 1e^{-3}$ with equal weights for variance and sparsity loss. In order to better generalize the attention module to a variety of content, we train a single attention model for all cover videos, with the training set containing 10 frames from each video sampled at regular intervals. The attention module was trained for 50 epochs. All experiments were performed on a single NVIDIA A100 GPU.

4.3. Main Results

Table 2 summarizes the average quantitative metrics of Peak Signal to Noise Ratio (PSNR) and SSIM for 5 cover videos with a given set of 128 x 128 hidden images. For each cover video, we report the PSNR and SSIM for the pretrained NeRV model in the top row, followed by the results obtained without and with attention. In the absence of attention, it was observed that for cover videos with high dynamic content, such as *Jockey* and *ShakeNDry*, the hidden image recovery quality was inferior as compared to those from relatively slow moving scenes. A sample of subjective results for this method are shown in Fig. 5. On the other hand, with attention, we notice for videos such as *Honeybee* and *ShakeNDry*, a greater quality is observed over almost all 4 measures as compared to our previous approach. Other than *Bosphorus*, introducing attention further enhances the fidelity of recovered hidden image at the **same or better quality** for the cover frames. This is generally hard to achieve with traditional steganography where we trade in steganographic frame quality in exchange for superior hidden recovery.

With the aid of the attention module, we further explore hiding images of larger dimensions, with sizes up to 1088 x 1088 (comparable to those of the cover frame). In Tab. 3 we report the results with *Beauty* cover video. Evidently, the reconstruction quality for hidden images drops significantly as we attempt to hide larger images. In Fig. 6, we highlight an example where the decoder could not recover high texture details precisely such as those along the terrace railings.



(a) Ground truth hidden image (b) Recovered hidden image

Figure 6. An example showing reconstruction artifacts (highlighted in yellow) when a 1088 x 1088 image is hidden using attention (cover video : *beauty*).

Table 3. Average quantitative metrics for StegaNeRV with attention evaluated with *Beauty* cover and varying resolutions of the hidden image.

Hidden Size	A. NeRV Output		B. Hidden Recovery	
	PSNR	SSIM	PSNR	SSIM
128 x 128	34.166	0.915	35.571	0.975
320 x 320	34.166	0.915	32.625	0.941
512 x 512	34.168	0.915	31.302	0.927
1088 x 1088	34.168	0.915	25.741	0.818

4.4. Case Study

We illustrate a common use case of video steganography where a creator wishes to embed ownership data (in this case, a logo) imperceptibly within the cover video without hindering the overall video quality. Hiding such information is desirable and in most instances, necessary to maintain authenticity and integrity of the content. In Fig. 7a, we show how the decoder H_ψ correctly reveals the hidden image when provided with a steganographic frame. In the second example, we attempt to recover the hidden image from the output of a pretrained NeRV *i.e.* one that has not been jointly trained with the particular steganographic decoder. As shown in Fig. 7b, such an attempt fails indicating that the secret image can only be reconstructed when a NeRV model is used in conjunction with its paired steganographic decoder.

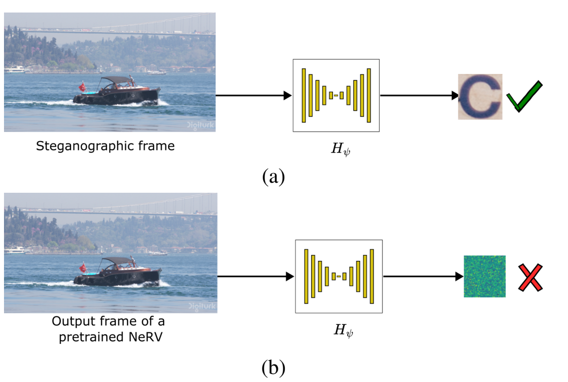


Figure 7. An illustration of a particular use case of ownership identification. (a) successful recovery by the decoder when a steganographic frame is given as input, while (b) no hidden image is recovered from an output frame of a pretrained NeRV.

5. Discussion and Future Work

Variable Gradient Scaling. As an extension of our work, we further explored if the hyper-parameter α, which decides the distribution of scaled gradients, can be made adaptive as per the influence of a given stage on the NeRV output. By virtue of the NeRV architecture, weight perturbations introduced near the output convolutional layers have greater potential for data hiding whereas such modifications are not desirable near the input, which operate on the embedded timestamp and thereby establish the video structure. To this end, we propose variable gradient scaling as per Eq. (9), where α_i increases progressively across 7 stages of NeRV (2 MLP layers + 5 NeRV blocks). Each stage has an associated state weight $s_i = i$, which is not a trainable parameter. We substitute α_i for the constant α while computing the gradient mask vector for the i^{th} stage, $\mathbf{c}_i$, as previously given in Eq. (4). We show a comparison of steganographic frame quality across training epochs in Fig. 8, where 320 x 320 images were hidden with *Beauty* cover video without attention. An improvement in steganographic frame quality is observed as compared to gradient scaling with constant α with all other parameters held constant. This shows by adapting α for different stages of NeRV, we can hide images with greater quality of the steganographic video.

$$\alpha_i = sin\left[\frac{\pi}{2}\left(\frac{\sum_{k=1}^{i} s_k}{\sum_{k=1}^{7} s_k}\right)\right] * scale \tag{9}$$

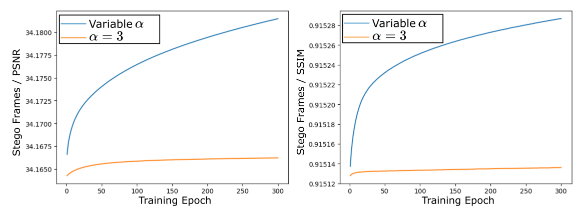


Figure 8. Improved PSNR (left) and SSIM (right) for steganographic frames with variable $\alpha \in (0, 3]$ (*i.e.* $scale = 3$), $s_i = i$ as compared to constant $\alpha = 3$. The hidden images are of size 320 x 320 with *Beauty* cover video.

Large-capacity Hiding. With our proposed architecture, hiding images with dimensions as large as the cover frame has not been realized with high precision. There are two key parts to this issue as the performance depends both on the hiding capacity of NeRV as well as on the ability of the decoder to recognize the subtle modifications in the steganographic frame for an accurate recovery. This would be our next step investigating large scale data hiding within NeRV.

6. Conclusion

This paper addresses the open problem of hiding data imperceptibly with its robust recovery within video frames represented by an implicit neural representation. We describe how jointly training the INR with the steganographic decoder is essential for accurately recovering hidden images without distorting the steganographic frames. Moreover, an attention-based approach further enhances hiding capacity by guiding the framework to emphasize hiding within rich texture regions. Our extensive experiments show promising results, prompting future research efforts in this area.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017. 6

[2] Yunpeng Bai, Chao Dong, Cairong Wang, and Chun Yuan. Ps-nerv: Patch-wise stylized neural representations for videos. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 41–45. IEEE, 2023. 1

[3] Shumeet Baluja. Hiding images in plain sight: Deep steganography. *Advances in Neural Information Processing Systems*, 30:2069–2079, 2017. 2

[4] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. 1

[5] Huanhuan Chai, Zhaohong Li, Fan Li, and Zhenzhen Zhang. An end-to-end video steganography network based on a coding unit mask. *Electronics*, 11(7):1142, 2022. 3

[6] Hao Chen. Nerv : Official pytorch implementation for video neural representation. https://github.com/haochen-rye/NeRV/commit/6904589ffbe8603dd742d9301bc1954abc551830, 2022. 6

[7] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. *Advances in Neural Information Processing Systems*, 34:21557–21568, 2021. 1, 2, 3, 7

[8] Lifeng Chen, Jia Liu, Yan Ke, Wenquan Sun, Weina Dong, and Xiaozhong Pan. Marknerf: Watermarking for neural radiance field. *arXiv preprint arXiv:2309.11747*, 2023. 2

[9] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021. 1

[10] Zeyuan Chen, Yinbo Chen, Jingwen Liu, Xingqian Xu, Vidit Goel, Zhangyang Wang, Humphrey Shi, and Xiaolong Wang. Videoinr: Learning video implicit neural representation for continuous space-time super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2047–2057, 2022. 3

[11] Daniel L Currie and Cynthia E Irvine. Surmounting the effects of lossy compression on steganography. In *Proceedings of the 19th National Information Systems Security Conference*, pages 194–201, 1996. 5

[12] Weina Dong, Jia Liu, Yan Ke, Lifeng Chen, Wenquan Sun, and Xiaozhong Pan. Steganography for neural radiance fields by backdooring. *arXiv preprint arXiv:2309.10503*, 2023. 2

[13] Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14304–14314. IEEE Computer Society, 2021. 3

[14] Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123*, 2021. 2

[15] Mahdi Hashemzadeh. Hiding information in videos using motion clues of feature points. *Computers & Electrical Engineering*, 68:14–25, 2018. 5

[16] Bo He, Xitong Yang, Hanyu Wang, Zuxuan Wu, Hao Chen, Shuaiyi Huang, Yixuan Ren, Ser-Nam Lim, and Abhinav Shrivastava. Towards scalable neural representation for diverse videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6132–6142, 2023. 3

[17] Vojtěch Holub and Jessica Fridrich. Digital image steganography using universal distortion. In *Proceedings of the first ACM workshop on Information hiding and multimedia security*, pages 59–68, 2013. 2

[18] Jaspreet Kaur and Jagroop Kaur. Hiding text in video using steganographic technique-a review. *Research Cell: An International Journal of Engineering Sciences*, pages 578–582, 2016. 1

[19] V Kavitha and KS Easwarakumar. Neural based steganography. In *PRICAI 2004: Trends in Artificial Intelligence: 8th Pacific Rim International Conference on Artificial Intelligence, Auckland, New Zealand, August 9-13, 2004. Proceedings 8*, pages 429–435. Springer, 2004. 2

[20] Imran Khan, Bhupendra Verma, Vijay K. Chaudhari, and Ilyas Khan. Neural network based steganography algorithm for still images. In *INTERACT-2010*, pages 46–51, 2010. 2

[21] Jaechang Kim, Yunjoo Lee, Seunghoon Hong, and Jungseul Ok. Learning continuous representation of audio for arbitrary scale super resolution. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3703–3707. IEEE, 2022. 1

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[23] Vinita V Korgaonkar and Manisha Naik Gaonkar. A dwt-dct combined approach for video steganography. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pages 421–424. IEEE, 2017. 2

[24] Jayakanth Kunhoth, Nandhini Subramanian, Somaya Al-Maadeed, and Ahmed Bouridane. Video steganography: recent advances and challenges. *Multimedia Tools and Applications*, 82(27):41943–41985, 2023. 1

[25] Luca A Lanzendörfer and Roger Wattenhofer. Siamese siren: Audio compression with implicit neural representations. *arXiv preprint arXiv:2306.12957*, 2023. 2

[26] Joo Chan Lee, Daniel Rho, Jong Hwan Ko, and Eunbyung Park. Ffnerv: Flow-guided frame-wise neural representations for videos. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7859–7870, 2023. 1

[27] Chenxin Li, Brandon Y Feng, Zhiwen Fan, Panwang Pan, and Zhangyang Wang. Steganerf: Embedding invisible information within neural radiance fields. In *Proceedings of*

the IEEE/CVF International Conference on Computer Vision, pages 441–453, 2023. 2, 3

[28] Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. *Advances in Neural Information Processing Systems*, 34:18114–18125, 2021. 1

[29] Zizhang Li, Mengmeng Wang, Huaijin Pi, Kechun Xu, Jianbiao Mei, and Yong Liu. E-nerv: Expedite neural video representation with disentangled spatial-temporal context. In *European Conference on Computer Vision*, pages 267–284. Springer, 2022. 1, 3

[30] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 1

[31] Shuyang Liu and Degang Xu. A robust steganography method for hevc based on secret sharing. *Cognitive Systems Research*, 59:207–220, 2020. 3

[32] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11006–11015, 2019. 1

[33] Shao-Ping Lu, Rong Wang, Tao Zhong, and Paul L Rosin. Large-capacity image steganography based on invertible neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10816–10825, 2021. 3

[34] Ziyuan Luo, Qing Guo, Ka Chun Cheung, Simon See, and Renjie Wan. Copyrnerf: Protecting the copyright of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22401–22411, 2023. 2

[35] Long Mai and Feng Liu. Motion-adjustable neural implicit video representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10738–10747, 2022. 2

[36] Alexandre Mercat, Marko Viitanen, and Jarno Vanne. Uvg dataset: 50/120fps 4k sequences for video codec analysis and development. In *Proceedings of the 11th ACM Multimedia Systems Conference*, pages 297–302, 2020. 6

[37] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 2

[38] J. Mielikainen. Lsb matching revisited. *IEEE Signal Processing Letters*, 13(5):285–287, 2006. 2

[39] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[40] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16190–16199, 2022. 1

[41] Aayush Mishra, Suraj Kumar, Aditya Nigam, and Saiful Islam. Vstegnet: Video steganography network using spatio-temporal features and micro-bottleneck. In *Proceedings of the British Machine Vision Conference (BMVC)*, page 274, 2019. 2

[42] Chong Mou, Youmin Xu, Jiechong Song, Chen Zhao, Bernard Ghanem, and Jian Zhang. Large-capacity and flexible video steganography via invertible neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22606–22615, 2023. 3

[43] Ramadhan J. Mstafa and Khaled M. Elleithy. A novel video steganography algorithm in dct domain based on hamming and bch codes. In *2016 IEEE 37th Sarnoff Symposium*, pages 208–213, 2016. 3

[44] Ramadhan J Mstafa, Khaled M Elleithy, and Eman Abdelfattah. A robust and secure video steganography method in dwt-dct domains based on multiple object tracking and ecc. *IEEE access*, 5:5354–5365, 2017. 3

[45] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 1

[46] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020. 2

[47] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 1

[48] Mritha Ramalingam and Nor Ashidi Mat Isa. Video steganography based on integer haar wavelet transforms for secured data transfer. *Indian Journal of Science and Technology*, 7(7):897–904, 2014. 2

[49] Mennatallah M Sadek, Amal S Khalifa, and Mostafa GM Mostafa. Video steganography: a comprehensive review. *Multimedia tools and applications*, 74:7063–7094, 2015. 1

[50] Bipasha Sen, Aditya Agarwal, Vinay P Namboodiri, and C.V. Jawahar. INR-v: A continuous representation space for video-based generative tasks. *Transactions on Machine Learning Research*, 2022. 3

[51] Liyue Shen, John Pauly, and Lei Xing. Nerp: Implicit neural representation learning with prior embedding for sparsely sampled image reconstruction. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):770–782, 2024. 1

[52] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020. 1

[53] Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit neural representations for image compression. In *European Conference on Computer Vision*, pages 74–91. Springer, 2022. 1, 2

[54] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video

coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012. 1, 3

[55] Meenu Suresh and I Shatheesh Sam. High secure video steganography based on shuffling of data on least significant dct coefficients. In *2018 Second international conference on intelligent computing and control systems (ICICCS)*, pages 877–882. IEEE, 2018. 2

[56] A Swathi and SAK Jilani. Video steganography by lsb substitution using different polynomial equations. *International Journal of Computational Engineering Research*, 2 (5):1620–1623, 2012. 2, 3

[57] Filip Szatkowski, Karol J Piczak, Przemysław Spurek, Jacek Tabor, and Tomasz Trzciński. Hypersound: Generating implicit neural representations of audio signals with hypernetworks. *arXiv preprint arXiv:2211.01839*, 2022. 2

[58] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021. 1, 2

[59] Yiqi Tew and KokSheik Wong. Information hiding in hevc standard using adaptive coding block size decision. In *2014 IEEE international conference on image processing (ICIP)*, pages 5502–5506. IEEE, 2014. 3

[60] Vandana Thakur and Monjul Saikia. Hiding secret image in video. In *2013 International Conference on Intelligent Systems and Signal Processing (ISSP)*, pages 150–153. IEEE, 2013. 1

[61] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017. 6

[62] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 2

[63] Xinyu Weng, Yongzhi Li, Lu Chi, and Yadong Mu. High-capacity convolutional video steganography with temporal residual modeling. In *Proceedings of the 2019 on international conference on multimedia retrieval*, pages 87–95, 2019. 2, 3

[64] Pin Wu, Xuting Chang, Yang Yang, and Xiaoqiang Li. Basn—learning steganography with a binary attention mechanism. *Future Internet*, 12(3):43, 2020. 5

[65] Mehdi Yedroudj, Frédéric Comby, and Marc Chaumont. Steganography using a 3-player game. *Journal of Visual Communication and Image Representation*, 72:102910, 2020. 3

[66] Zeyad Safaa Younus and Ghada Thanoon Younus. Video steganography using knight tour algorithm and lsb method for encrypted data. *Journal of Intelligent Systems*, 29(1): 1216–1225, 2019. 3

[67] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 1

[68] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. In *International Conference on Learning Representations*, 2022. 3

[69] Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust invisible video watermarking with attention. *arXiv preprint arXiv:1909.01285*, 2019. 3

[70] Ru Zhang, Shiqi Dong, and Jianyi Liu. Invisible steganography via generative adversarial networks. *Multimedia tools and applications*, 78(7):8559–8575, 2019. 3

[71] Yunfan Zhang, Ties van Rozendaal, Johann Brehmer, Markus Nagel, and Taco Cohen. Implicit neural video compression. *arXiv preprint arXiv:2112.11312*, 2021. 1, 2

[72] Qi Zhao, M Salman Asif, and Zhan Ma. Dnerv: Modeling inherent dynamics via difference neural representation for videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2031–2040, 2023. 1