Large-capacity and Flexible Video Steganography via Invertible Neural Network

Chong Mou[1], Youmin Xu[1], Jiechong Song[1], Chen Zhao[2], Bernard Ghanem[2], Jian Zhang[1,3,*]
[1]Peking University Shenzhen Graduate School, Shenzhen, China
[2]King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia
[3]Peng Cheng Laboratory, Shenzhen, China
{eechongm, youmin.xu}@stu.pku.edu.cn; {songjiechong, zhangjian.sz}@pku.edu.cn;
{chen.zhao, Bernard.Ghanem}@kaust.edu.sa

Abstract

*Video steganography is the art of unobtrusively concealing secret data in a cover video and then recovering the secret data through a decoding protocol at the receiver end. Although several attempts have been made, most of them are limited to low-capacity and fixed steganography. To rectify these weaknesses, we propose a **L**arge-capacity and **F**lexible **V**ideo **S**teganography **N**etwork (LF-VSN) in this paper. For large-capacity, we present a reversible pipeline to perform multiple videos hiding and recovering through a single invertible neural network (INN). Our method can **hide/recover 7 secret videos in/from 1 cover video** with promising performance. For flexibility, we propose a key-controllable scheme, enabling different receivers to recover particular secret videos from the same cover video through specific keys. Moreover, we further improve the flexibility by proposing a scalable strategy in multiple videos hiding, which can hide variable numbers of secret videos in a cover video with a single model and a single training session. Extensive experiments demonstrate that with the significant improvement of the video steganography performance, our proposed LF-VSN has high security, large hiding capacity, and flexibility. The source code is available at https://github.com/MC-E/LF-VSN.*

1. Introduction

Steganography [10] is the technology of hiding some secret data into an inconspicuous cover medium to generate a stego output, which only allows the authorized receiver to recover the secret information. Unauthorized people can only access the content of the plain cover medium, and hard to detect the existence of secret data. In the current digital world, image and video are commonly used covers, widely applied in digital communication [27], copyright protection [36], information certification [31], e-commerce [26], and many other practical fields [10, 12].

Traditional video steganography methods usually hide messages in the spatial domain or transform domain by manual design. Video steganography in the spatial domain means embedding is done directly to the pixel values of video frames. Least significant bits (LSB) [8, 45] is the most well-known spatial-domain method, replacing the n least significant bits of the cover image with the most significant n bits of the secret data. Many researchers have used LSB replacement [6] and LSB matching [34] for video steganography. The transform-domain hiding [5, 17, 39] is done by modifying certain frequency coefficients of the transformed frames. For instance, [44] proposed a video steganography technique by manipulating the quantized coefficients of DCT (Discrete Cosine Transformation). [9] proposed to compare the DWT (Discrete Wavelet Transformation) coefficients of the secret image and the cover video for hiding. However, these traditional methods have low hiding capacity and invisibility, easily being cracked by steganalysis methods [15, 28, 33].

Recently, some deep-learning methods were proposed to improve the hiding capacity and performance. Early works are presented in image steganography. Baluja [3, 4] proposed the first deep-learning method to hide a full-size image into another image. Recently, [21, 32] proposed designing the steganography model as an invertible neural network (INN) [13, 14] to perform image hiding and recovering with a single model. For video steganography, Khare et al. [22] first utilized back propagation neural networks to improve the performance of the LSB-based scheme. [43] is the first deep-learning method to hide a video into another video. Unfortunately, it simply aims to hide the residual across adjacent frames in a frame-by-frame manner, and it requires several separate steps to complete the video hiding and re-

[*] *Corresponding author*. This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research through the Visual Computing Center (VCC) funding, SDAIA-KAUST Center of Excellence in Data Science and Artificial Intelligence, and Shenzhen Research Project JCYJ20220531093215035.

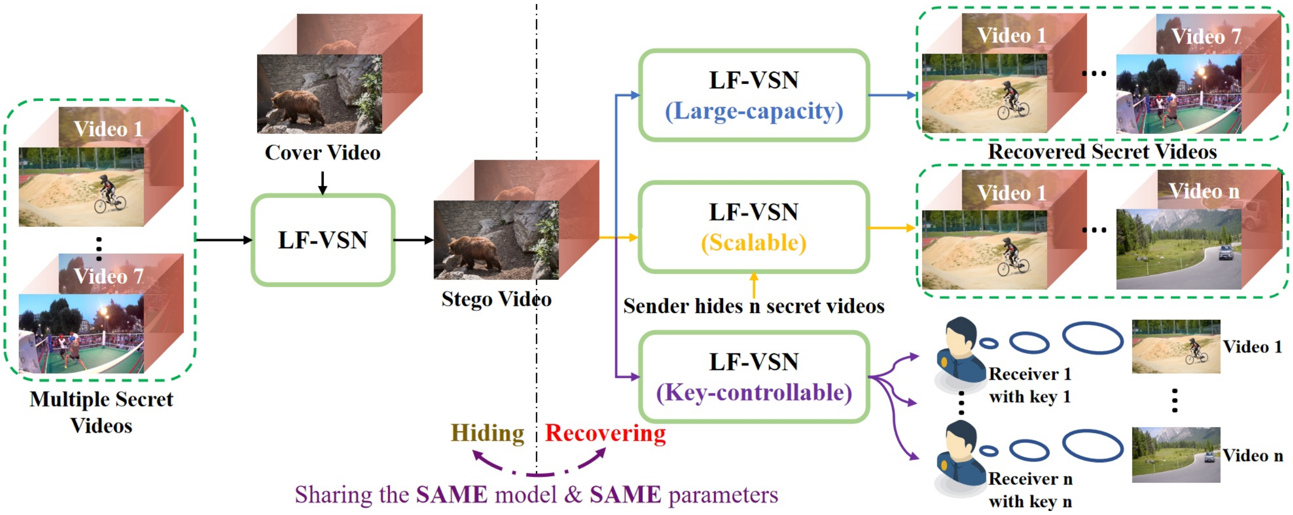


Figure 1. Illustration of our large-capacity and flexible video steganography network (LF-VSN). Our LF-VSN reversibly solves multiple videos hiding and recovering with a single model and the same parameters. It has large-capacity, key-controllable and scalable advantages.

covering. [35] utilize 3D-CNN to explore the temporal correlation in video hiding. However, it utilizes two separated 3D UNet to perform hiding and recovering, and it has high model complexity (367.2 million parameters). While video steganography has achieved impressive success in terms of hiding capacity to hide a full-size video, the more challenging multiple videos hiding has hardly been studied. Also, the steganography pipeline is rigid.

In this paper, we study the large-capacity and flexible video steganography, as shown in Fig. 1. Concretely, we propose a reversible video steganography pipeline, achieving large capacity to hide/recover multiple secret videos in/from a cover video. At the same time, our model complexity is also attractive by combining several weight-sharing designs. The flexibility of our method is twofold. First, we propose a key-controllable scheme, enabling different receivers to recover particular secret videos with specific keys. Second, we propose a scalable strategy, which can hide variable numbers of secret videos into a cover video with a single model and a single training session. To summarize, this work has the following contributions:

- We propose a large-capacity video steganography method, which can hide/recover multiple (**up to 7**) secret videos in/from a cover video. Our hiding and recovering are fully reversible via a single INN.
- We propose a key-controllable scheme with which different receivers can recover particular secret videos from the same cover video via specific keys.
- We propose a scalable embedding module, utilizing a single model and a single training session to satisfy different requirements for the number of secret videos hidden in a cover video.
- Extensive experiments demonstrate that our proposed method achieves state-of-the-art performance with large hiding capacity and flexibility.

2. Related Work

2.1. Video Steganography

Steganography can date back to the 15th century, whose goal is to encode a secret message in some transport mediums and covertly communicate with a potential receiver who knows the decoding protocol to recover the secret message. Since the human visual system is less sensitive to small changes in digital media, especially digital videos. Video steganography is becoming an important research area in various data-hiding technologies [10].

Traditional video steganography methods usually performed hiding and recovering in the spatial domain, *e.g.*, Pixel Value Differencing (PVD) [20, 40] and Least Significant Bits (LSB) [6, 34, 41]. PVD embeds the secret data in the difference value of two adjacent pixels. In [40], a PVD-based video steganography system is proposed to embed the secret data in a compressed domain of the cover medium. [20] utilized enhanced pixel-value differencing (EPVD) to improve the video steganography performance. LSB methods work by replacing the n least significant bits of the cover data with the most significant n bits of the secret information. [41] utilized LSB replacement technique to hide secret text in grayscale video frames. To enhance the security in LSB-based methods, [2] shuffled the secret data and embedded the index of correct order into the cover video. In addition to spatial-domain methods, some transformed domain methods [9, 44] were proposed to perform hiding by modifying certain frequency coefficients of the transformed cover video. For instance, [44] proposed a video steganography technique by manipulating the quantized coefficients of DCT transformation. [9] proposed to compare the DWT transformation coefficients of the secret image and the cover video for hiding. Nevertheless, the above traditional methods have low hiding capacity and invisibility, easily producing artificial markings and being cracked by steganalysis methods [15, 28, 33].

Motivated by the success of deep learning, some deep-learning methods were proposed. [16] introduced GAN to steganography, showing that the adversarial training scheme can improve hiding security. [49] improve the hiding quality by utilizing two independent adversarial networks to critique the video quality and optimize for robustness. [25] studied the lossless steganography below 3 bits per pixel (bpp) hiding. [38] embedded the secret data in the wavelet transform coefficients of the video frames. The above methods focus more on the robustness of low-capacity hiding. One of the important applications of low-capacity steganography is watermarking [1, 42, 52], in which the secret bit string represents the sign of the owner. Some deep-learning methods were proposed for large-capacity hiding. [3, 4] first explored hiding a full-size image into another image. [21, 32] proposed a cheaper pipeline by implementing image hiding and recovering with a single invertible neural network (INN) [13, 14]. Compared with image hiding, video hiding is a more challenging task, requiring a larger hiding capacity. [43] first studied to hide/recover a video in/from another video. However, this method simply hides the residual across adjacent frames in a frame-by-frame manner. [35] explores the temporal correlation by 3D CNN in video steganography. However, it utilizes two separate 3D UNet to perform hiding and recovering and has high model complexity ($367.2M$ model parameters). These previous works demonstrate that deep networks have great potential in video hiding, inspiring us to study the more challenging task of multiple and flexible video hiding.

2.2. Invertible Neural Nerwork

Since the concept of invertible neural network (INN) was proposed in [13, 14], INN has attracted more and more attention due to its pure invertible pipeline. Pioneering research on INN can be seen in image generation tasks. For instance, Glow [24] utilized INN to construct an invertible mapping between the latent variable $\mathbf{z}$ and nature images $\mathbf{x}$. Specifically, the generative process $\mathbf{x} = f_\theta(\mathbf{z})$ given a latent variable can be specified by an INN architecture f_θ. The direct access to the inverse mapping $\mathbf{z} = f_\theta^{-1}(\mathbf{x})$ makes inference much cheaper. Up to now, INN has been studied in several vision tasks (*e.g.*, image rescaling [19, 46], image restoration [29, 30], image coloring [51], and video temporal action localization [50]) and presents promising performance.

The architecture of INN needs to be carefully designed to guarantee the invertibility. Commonly, INN is composed of several invertible blocks, *e.g.*, the coupling layer [13]. Given the input $\mathbf{h}$, the coupling layer first splits $\mathbf{h}$ into two parts ($\mathbf{h}_1$ and $\mathbf{h}_2$) along the channel axis. Then they undergo the affine transformations with the affine parameters generated by each other:

$$\begin{aligned} \hat{\mathbf{h}}_1 &= \mathbf{h}_1 \cdot \psi_1(\mathbf{h}_2) + \phi_1(\mathbf{h}_2) \\ \hat{\mathbf{h}}_2 &= \mathbf{h}_2 \cdot \psi_2(\hat{\mathbf{h}}_1) + \phi_2(\hat{\mathbf{h}}_1), \end{aligned} \tag{1}$$

where $\psi(\cdot)$ and $\phi(\cdot)$ are arbitrary functions. $\hat{\mathbf{h}}_1$ and $\hat{\mathbf{h}}_2$ are the outputs of the coupling layer. Correspondingly, the inverse process is defined as:

$$\mathbf{h}_1 = \frac{\hat{\mathbf{h}}_1 - \phi_1(\mathbf{h}_2)}{\psi_1(\mathbf{h}_2)}; \qquad \mathbf{h}_2 = \frac{\hat{\mathbf{h}}_2 - \phi_2(\hat{\mathbf{h}}_1)}{\psi_2(\hat{\mathbf{h}}_1)}. \tag{2}$$

In this paper, we employ the reversible forward and backward processes of INN to perform multiple videos hiding and recovering, respectively. We further improve INN to explore flexible video steganography.

3. Methodology

3.1. Overview

An overview of our LF-VSN is presented in Fig. 2. Specifically, given N_s secret videos $\mathbf{x}_{se} = \{\mathbf{x}_{se}(n)\}_{n=1}^{N_s}$ and a cover video $\mathbf{x}_{co}$, the forward hiding is operated group-by-group through a sliding window, traversing each video from head to tail. After hiding, a stego video $\mathbf{x}_{st}$ is produced, ostensibly indistinguishable from $\mathbf{x}_{co}$ to ensure that $\mathbf{x}_{se}$ is undetectable. In the backward recovering, a channel-wise broadcasting operation ($\mathbb{R}^{3\times W\times H} \xrightarrow{copy} \mathbb{R}^{3L\times W\times H}$) copies each stego frame in the channel dimension to form the reversed input. During recovering, multiple secret videos are recovered frame-by-frame in parallel. It is worth noting that the forward hiding and backward recovering share the same model architecture and parameters.

3.2. Steganography Input and Output Design

At the beginning of each hiding step, a fusion module is applied to fuse frames in each group to take advantage of the inner temporal correlation. Considering that it is easy to produce texture artifacts and color distortion when hiding in the spatial dimension [15, 21], we perform the fusion by a frequency concatenation. Specifically, given the j-th cover group $\mathbf{X}_{co\circledast j} \in \mathbb{R}^{L\times 3\times W\times H}$ and secret groups $\{\mathbf{X}_{se\circledast j}(n) \in \mathbb{R}^{L\times 3\times W\times H}\}_{n=1}^{N_s}$ (each contains L frames), we adopt Haar discrete wavelet transform (DWT) to split each frame into four frequency bands (*i.e.*, LL, HL, LH, HH). In each frame group, we concatenate the part in the same frequency band from different frames in the channel dimension and then concatenate these four bands in order of frequency magnitude, producing the final secret input $\{\mathbf{X}_{se\circledast j}(n) \in \mathbb{R}^{12L_s\times\frac{W}{2}\times\frac{H}{2}}\}_{n=1}^{N_s}$ and cover input $\mathbf{X}_{co\circledast j} \in \mathbb{R}^{12L_c\times\frac{W}{2}\times\frac{H}{2}}$. The output of the forward

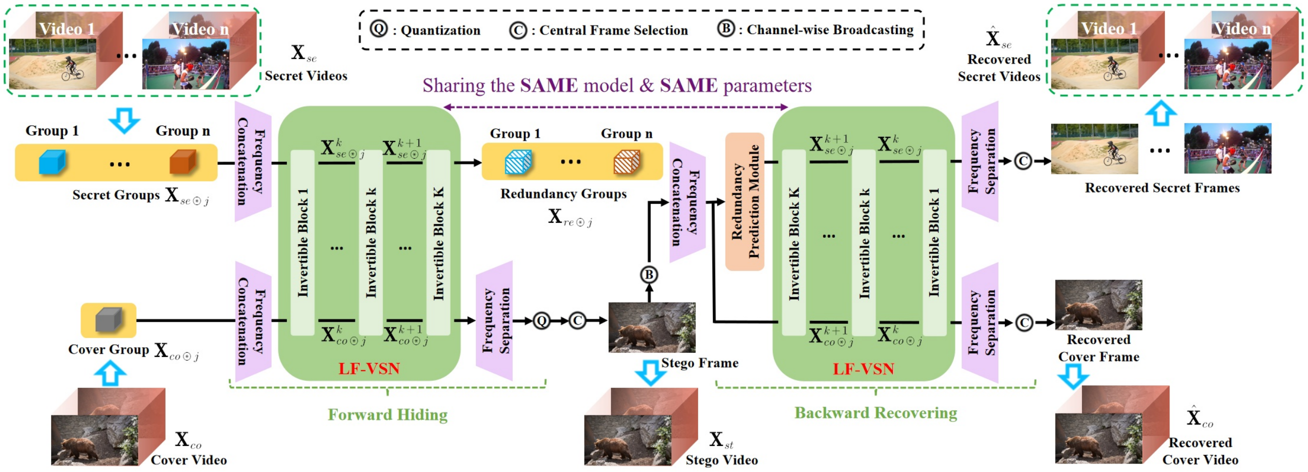


Figure 2. Network architecture of our LF-VSN. It is composed of several invertible blocks. In the forward hiding process, multiple secret videos are hidden in a cover video to generate a stego video, together with redundancy. In the backward recovering process, the stego video and predicted redundancy are fed to the reverse data flow of the same network with the same parameters to recover secret videos.

hiding comprises a stego group $\mathbf{X}_{st\circledast j}$ and several redundancy groups $\{\mathbf{X}_{re\circledast j}(n)\}_{n=1}^{N_s}$. $\mathbf{X}_{st\circledast j}$ is converted from the frequency domain to the spatial domain by a frequency separation, *i.e.*, the inverse of the frequency concatenation. $\mathbf{X}_{re\circledast j}(n)$ represents the redundancy of the $\mathbf{X}_{se\circledast j}(n)$ that does not need to be hidden and will be discarded. In our LF-VSN, we utilize the adjacent frames to cooperate with hiding the central frame. Thus, we only output the central stego frame in each hiding step. The backward recovering is similarly operated in the frequency domain and converted to the spatial domain at the output.

3.3. Invertible Block

As shown in Fig. 2, our hiding and recovering have reverse information flow constructed by several invertible blocks (IBs). The architecture of IB is presented in Fig. 3. Concretely, in the k-th IB, there are two branches to process the input cover group $\mathbf{X}^k_{co\circledast j}$ and secret groups $\{\mathbf{X}^k_{se\circledast j}(n)\}_{n=1}^{N_s}$, respectively. Several interaction pathways between these two branches construct the invertible projection. We use an additive transformation to project the cover branch and employ an enhanced affine transformation to project the secret branch. The transformation parameters are generated from each other. Here we utilize weight-sharing modules ($\eta_k^1(\cdot)$ and $\phi_k^1(\cdot)$) to extract features from all secret groups, producing a feature set $\{\mathbf{F}^k_{se}(n)\}_{n=1}^{N_s} = \{\phi_k(\eta_k(\mathbf{X}^k_{se\circledast j}(n)))\}_{n=1}^{N_s}$. $\eta_k^i(\cdot)$ and $\phi_k^i(\cdot)$ $(i = 1, 2, 3)$ refer to a 3×3 convolution layer and a five-layer dense block [18], respectively. Then, we concatenate $\mathbf{F}^k_{se}$ in the channel dimension and pass through an aggregation module $\xi_k(\cdot)$ to generate the transformation parameters of the cover branch. Note that $\xi_k(\cdot)$ is optional in different cases. In our fixed hiding, $\xi_k(\cdot)$ is a 3×3 convolution layer, and it is a scalable embedding module in our scalable hiding. The

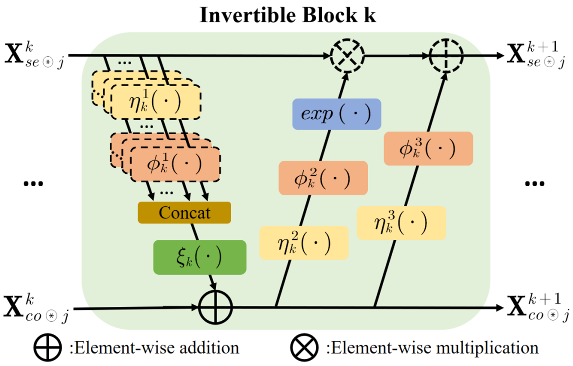


Figure 3. Illustration of the architecture of our invertible block. The dashed line refers to weight sharing.

transformation parameters of the secret branch are generated from $\mathbf{X}^k_{co\circledast j}$ and shared among different secret groups. Thus, in the k-th invertible block, the bijection of the forward propagation in Eq. (1) is reformulated as:

$$\begin{aligned} &\mathbf{X}^{k+1}_{co\circledast j} = \mathbf{X}^k_{co\circledast j} + \xi_k(||\phi_k^1(\eta_k^1(\mathbf{X}^k_{se\circledast j}(n)))||_{n=1}^{N_s}) \\ &\{\mathbf{X}^{k+1}_{se\circledast j}(n)\}_{n=1}^{N_s} = \\ &\mathbf{X}^k_{se\circledast j}(n) \cdot \exp(\phi_k^2(\eta_k^2(\mathbf{X}^{k+1}_{co\circledast j}))) + \phi_k^3(\eta_k^3(\mathbf{X}^{k+1}_{co\circledast j})), \end{aligned} \tag{3}$$

where $||\cdot||$ refers to the channel-wise concatenation. $\exp(\cdot)$ is the Exponential function. Accordingly, the backward propagation is defined as:

$$\begin{aligned} &\{\mathbf{X}^k_{se\circledast j}(n)\}_{n=1}^{N_s} = \\ &(\mathbf{X}^{k+1}_{se\circledast j}(n) - \phi_k^3(\eta_k^3(\mathbf{X}^{k+1}_{co\circledast j}))) \cdot \exp(-\phi_k^2(\eta_k^2(\mathbf{X}^{k+1}_{co\circledast j}))) \\ &\mathbf{X}^k_{co\circledast j} = \mathbf{X}^{k+1}_{co\circledast j} - \xi_k(||\phi_k^1(\eta_k^1(\mathbf{X}^k_{se\circledast j}(n)))||_{n=1}^{N_s}). \end{aligned} \tag{4}$$

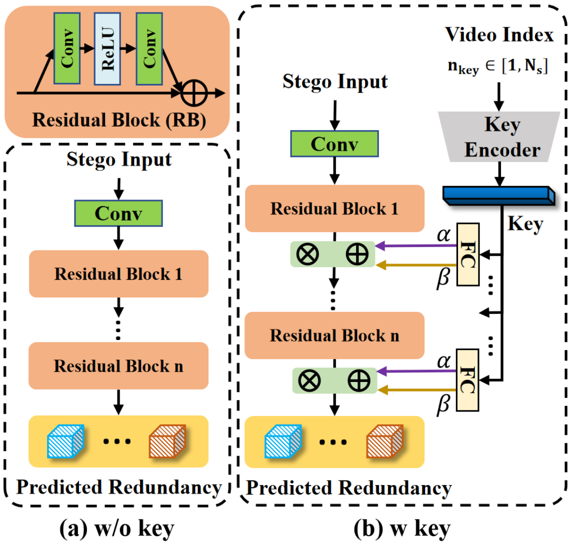


Figure 4. The architecture of our redundancy prediction module (RPM). It has two model settings: (a) RPM without (w/o) key controlling; (b) RPM with (w) key controlling.

3.4. Redundancy Prediction Module (RPM) & Key-controllable Design

As illustrated previously, we retain the stego part and discard the redundancy information in the forward hiding. Therefore, we need to prepare a suitable redundancy in the backward process to utilize the reversibility of INN to reconstruct the forward input (*i.e.*, secret and cover). In different tasks, most INN-based methods [21,24,32,46] constrain the generated redundancy information to obey the Gaussian distribution and utilize random Gaussian sampling to approximate this part in the backward process. Nevertheless, such random sampling lacks data specificity and adaptivity. In our LF-VSN, we predict the redundancy information from the stego group through a redundancy prediction module (RPM), as shown in Fig. 4(a). It is composed of several residual blocks (RB) without the Batch Normalization layer.

In this paper, we present a novel extension of RPM to construct key-controllable video steganography, with which we can hide multiple secret videos in a cover video and recover a secret video conditioned on a specific key. The architecture is shown in Fig. 4(b). Given the index n_{key} of a secret video $\mathbf{X}_{se}(n_{key})$, a specific key is generated by a key encoder, which is composed of several fully connected (FC) layers. The key is then fed into a FC layer at the end of each RB in RPM to generate a condition vector with $2C_{rpm}$ channels, which is divided into two modulation vectors $\alpha, \beta \in \mathbb{R}^{C_{rpm}\times 1\times 1}$ in the channel dimension. C_{rpm} is the feature channel of each RB in RPM. Then we modulate the output feature $\mathbf{F}_{rpm}$ of each RB as $\mathbf{F}_{rpm}\cdot\alpha+\beta$. In the training process, we constrain the recovered output the same as the n_{key}-th secret video (*i.e.*, $\mathbf{X}_{se}(n_{key})$). More details can be found in Sec. 3.6.

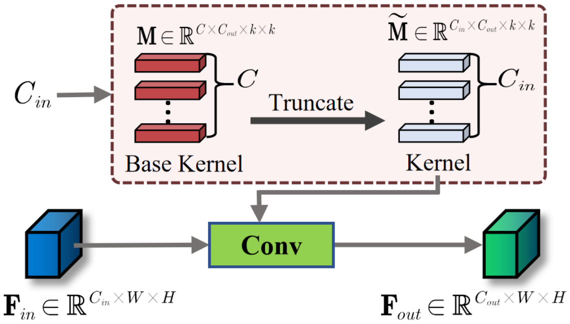

$\mathbf{M} \in \mathbb{R}^{C\times C_{out}\times k\times k}$
$\widetilde{\mathbf{M}} \in \mathbb{R}^{C_{in}\times C_{out}\times k\times k}$
C_{in}
C
Truncate
C_{in}
Base Kernel
Kernel
Conv
$\mathbf{F}_{in} \in \mathbb{R}^{C_{in}\times W\times H}$
$\mathbf{F}_{out} \in \mathbb{R}^{C_{out}\times W\times H}$

Figure 5. Illustration of our scalable embedding module. It takes the input feature map with scalable channels $C_{in} \in [1, C]$ and produces output features with fixed channels C_{out}.

3.5. Scalable Embedding Module

The scalable design is used to handle the case where there are different requirements for the number of secret videos hidden in a cover video. It is succinctly designed on the feature aggregation part $\xi_k(\cdot)$ in each IB, as shown in Fig. 3. The illustration of our scalable embedding module is presented in Fig. 5. It can be regarded as a special convolution layer, whose dimension of the convolution kernel can be changed according to the input. All convolution kernels $\widetilde{\mathbf{M}}$ with different dimensions are parameter-shared from the same base kernel $\mathbf{M}$. Technically, given the input feature $\mathbf{F}_{in} \in \mathbb{R}^{C_{in}\times W\times H}$, we truncate a convolution kernel $\widetilde{\mathbf{M}} \in \mathbb{R}^{C_{in}\times C_{out}\times k\times k}$ from $\mathbf{M} \in \mathbb{R}^{C\times C_{out}\times k\times k}$ to match the input dimension and then perform convolution: $\mathbf{F}_{out} = \widetilde{\mathbf{M}} * \mathbf{F}_{in}$. In this way, the training of $\mathbf{M}$ is completed through the training of all sub-kernels $\widetilde{\mathbf{M}}$.

3.6. Loss Function

In our LF-VSN, the loss function is used to constrain two parts, *i.e.*, forward hiding and backward recovering. The forward hiding is to hide multiple secret videos in the cover video. The generated stego video $\mathbf{X}_{st}$ should be undetectable to secret videos and as similar as possible to the cover video. Therefore, we constrain $\mathbf{X}_{st}$ to be the same as the cover video $\mathbf{X}_{co}$:

$$\mathcal{L}_f = ||\mathbf{X}_{st\circledast j}[I_c] - \mathbf{X}_{co\circledast j}[I_c]||_2^2, \tag{5}$$

where $||\cdot||_2^2$ donates the ℓ_2 norm. I_c is the index of the central frame in each group. In the backward recovering, there are two patterns: with and without key controlling. In both patterns, we aim to recover the secret information from the cover video. The difference stands between recovering a specific secret video and all secret videos. In the pattern without key controlling, the loss function is defined as:

$$\mathcal{L}_b = \sum_{n=1}^{N_s} ||\hat{\mathbf{X}}_{se\circledast j}(n)[I_c] - \mathbf{X}_{se\circledast j}(n)[I_c]||_2^2 + \\ ||\hat{\mathbf{X}}_{co\circledast j}[I_c] - \mathbf{X}_{co\circledast j}[I_c]||_2^2, \tag{6}$$

Table 1. Quantitative comparison (PSNR/SSIM) on Vimeo-T200. The best and second-best results are **highlighted** and underlined. Our LF-VSN achieves the best performance in stego and secret quality with acceptable complexity.

	Weng *et al.* [43]	Baluja [4]	ISN [32]	HiNet [21]	RIIS [47]	PIH [11]	LF-VSN (Ours)
Stego	29.43/0.862	34.14/0.860	42.08/0.965	42.09/0.962	43.50/0.951	-	**45.17/0.980**
Secret	32.08/0.899	35.21/0.931	42.11/0.984	44.44/0.991	44.08/0.964	36.48/0.939	**48.39/0.996**
Params.	42.57M	2.65M	3.00M	4.05M	8.15M	**0.67**M	7.40M

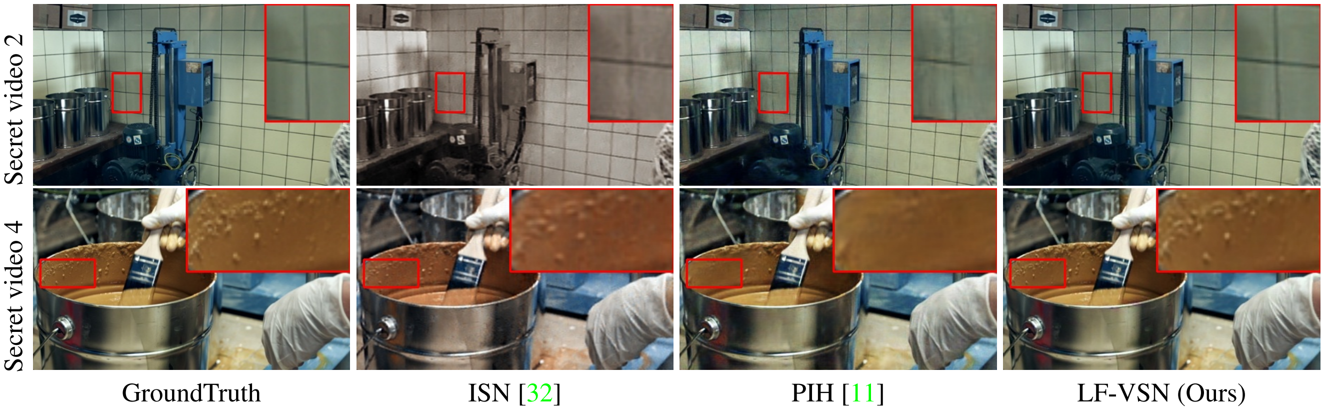


Figure 6. Visual comparison between our LF-VSN, ISN [32], and PIH [11] in 4 videos Steganography. We present the secret reconstruction results of video 2 and video 4. Our LF-VSN produces better result with intact color and details.

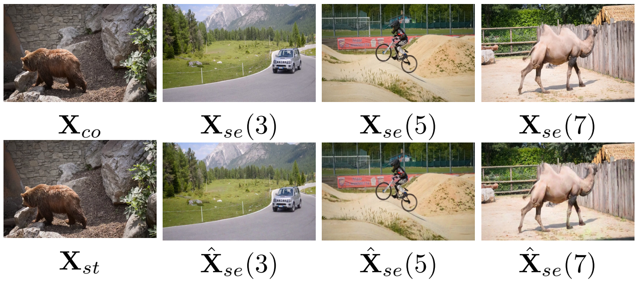


Figure 7. Visualization of our LF-VSN in 7 videos steganography, showing promising performance in such an extreme case.

where $\hat{\mathbf{X}}_{se}$ and $\hat{\mathbf{X}}_{co}$ represent the recovered secret and cover videos. In the pattern with key controlling, the loss function is defined to guarantee that the key generated from the video index n_{key} can only recover the n_{key}-th secret video. Thus, the loss function is reformulated as:

$$\mathcal{L}_b = ||\frac{1}{N_s}\sum_{n=1}^{N_s}\hat{\mathbf{X}}_{se\circledast j}(n)[I_c] - \mathbf{X}_{se\circledast j}(n_{key})[I_c]||_2^2 + ||\hat{\mathbf{X}}_{co\circledast j}[I_c] - \mathbf{X}_{co\circledast j}[I_c]||_2^2. \tag{7}$$

We optimize our LF-VSN by minimizing the forward loss function $\mathcal{L}_f$ and backward loss function $\mathcal{L}_b$ as:

$$\mathcal{L} = \mathcal{L}_f + \lambda\mathcal{L}_b, \tag{8}$$

where λ is a hyper-parameter to make a trade-off between forward hiding and backward recovering. We set $\lambda = 4$ to balance these two loss portions.

4. Experiment

4.1. Implementation Details

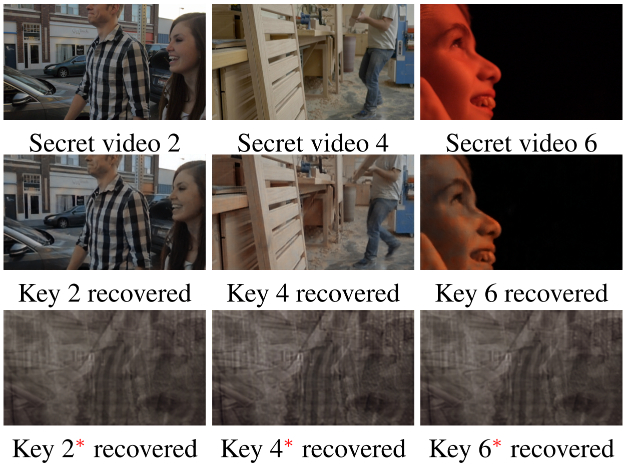
In this work, we adopt the training set of Vimeo-90K [48] to train our LF-VSN. Each sequence has a fixed spatial resolution of 448×256. During training, we randomly crop training videos to 144×144 with random horizontal and vertical flipping to make a data augmentation. We use Adam optimizer [23], with $\beta_1 = 0.9$, $\beta_2 = 0.5$. We set the batch size as 16. The weight decay factor is set as 1×10^{-12}. We use an initial learning rate of 1×10^{-4}, which will decrease by half for every $30K$ iterations. The number of total iterations is set as $250K$. The training process can be completed on one NVIDIA Tesla V100 GPU within 3 days. For testing, we select 200 sequences from the testing set of Vimeo-90K, denoted as Vimeo-T200 in this paper.

4.2. Comparison Against Other Methods

Here we compare our LF-VSN with other methods on single video steganography and challenging multiple videos steganography. The evaluation includes the stego quality in forward hiding and the secret quality in backward recovering. For single video steganography, we compare our LF-VSN with some well-known methods [4, 43] and recent proposed methods [11, 21, 32, 47]. Note that PIH [11] highlighted the need to quantize the stego image from the floating-point format of 32×3 to 8×3 bits per pixel. But PIH just added the quantization to the compared methods without retraining. Here we retrain HiNet [21] with

Table 2. Multiple videos steganography comparison (PSNR) of our LF-VSN, ISN [32], and PIH [11] on Vimeo-T200 test set. Our LF-VSN can hide/recover 7 videos with promising performance.

	Videos	2	3	4	5	6	7
ISN	Stego	37.60	36.41	32.56	31.46	-	-
	Secret	41.47	38.76	33.42	33.39	-	-
PIH	Stego	-	-	-	-	-	-
	Secret	35.95	34.96	34.20	-	-	-
Ours	Stego	**40.97**	**38.55**	**37.55**	**36.57**	**35.68**	**35.01**
	Secret	**44.24**	**42.27**	**40.21**	**38.88**	**36.94**	**35.71**



Secret video 2 Secret video 4 Secret video 6

Key 2 recovered Key 4 recovered Key 6 recovered

Key 2* recovered Key 4* recovered Key 6* recovered

Figure 8. Visualization of our key-controllable scheme in 6 videos steganography. In the second and third rows, we use the correct and wrong (*) keys of 2, 4, 6 to recover secret videos, respectively.

quantization to make a more fair comparison. Thus, its performance may be slightly higher than that reported in PIH. ISN [32], RIIS [47] and PIH were originally designed with quantization, which can be directly compared. Tab. 1 presents that our method achieves the best performance on stego and secret while maintaining acceptable complexity.

For multiple videos steganography, ISN [32] and PIH [11] studied how to hide multiple secret images in a cover image, which can be competitive counterparts of our LF-VSN. ISN can hide up to 5 secret images into 1 cover image, and PIH can hide 4 secret images. The comparison in Tab. 3 shows the better performance of our LF-VSN. Even in the 7 videos hiding, our method still has promising stego and secret quality ($> 35dB$). We present the visual comparison of different methods in 4 videos steganography in Fig. 6. Obviously, ISN has color distortion, and PIH has a loss of details. By contrast, our LF-VSN can recover high-fidelity results. We also present the secret and stego quality of our LF-VSN in 7 videos hiding in Fig. 7. These videos are from DAVIS [37] dataset. One can see that our LF-VSN has promising performance in such an extreme case.

4.3. Key-controllable Video Steganography

Hiding multiple secret videos in a cover video is challenging; doing so for different receivers is even more difficult. In this paper, we present a key-controllable scheme in multiple videos steganography. It enables different receivers to recover particular secret videos through specific keys. The comparison in Tab. 3 presents that our controllable scheme still has a large hiding capacity (up to 6 videos) with attractive performance ($> 30dB$). The visualization of recovering quality is presented in the second row of Fig. 8, showing the high-quality and key-controllable results of our LF-VSN in multiple videos steganography.

Table 3. Performance comparison between controllable (C) and non-controllable (NC) video steganography of our LF-VSN.

Num. videos	2	4	6
Stego (NC/C)	40.97/38.67	37.55/34.41	35.68/30.48
Secret (NC/C)	44.24/41.04	40.21/37.15	36.94/31.95

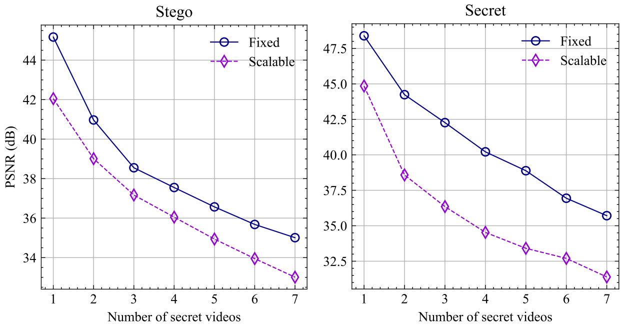


Figure 9. Performance comparison between our scalable and fixed design in multiple videos steganography.

We also study the security of our controlling scheme, *i.e.*, the key is sensitive and model-specific. Here we take two sets of parameters, producing from the 250K and 240K iterations in the same training process. We use the key produced by one model (*) to recover the secret video hidden by another. The result in the third row of Fig. 8 presents that the wrong key has no controlling and recovering ability. Thus, our key-controllable scheme not only has the controlling function but also enhances data security.

4.4. Scalable Video Steganography

In this paper, we present a scalable scheme in multiple videos steganography. It can hide a variable number of secret videos into a cover video with a single model. We evaluate the performance of our scalable design and compare it with the fixed version in Tab. 9. Obviously, our method has an attractive performance ($> 31dB$) in hiding a variable number (up to 7) of secret videos into a cover video by a single model. The performance degradation compared to fixed version is acceptable. With this design, a single model can satisfy multiple steganography demands.

4.5. Steganographic Analysis

The data security is one of the most important concerns in steganography. In this section, we evaluate the anti-

Table 4. The ablation study of different components in our LF-VSN. It includes the sliding window size, number of invertible blocks (IB), frequency concatenation (FreqCat), and redundancy prediction module (RPM).

Num. videos	2			4			6			3			3	3
Window size	1	3	5	1	3	5	1	3	5	3	3 (ours)	3	3	3
Num. IB	16	16	16	16	16	16	16	16	16	12	16	20	16	16
FreqCat	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
RPM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓
Stego	39.64	40.97	41.08	36.41	37.55	37.86	34.47	35.46	35.96	38.03	38.55	38.91	38.28	36.85
Secret	42.97	44.24	44.43	37.67	40.21	40.42	35.11	36.83	39.97	41.99	42.27	42.40	41.69	40.36

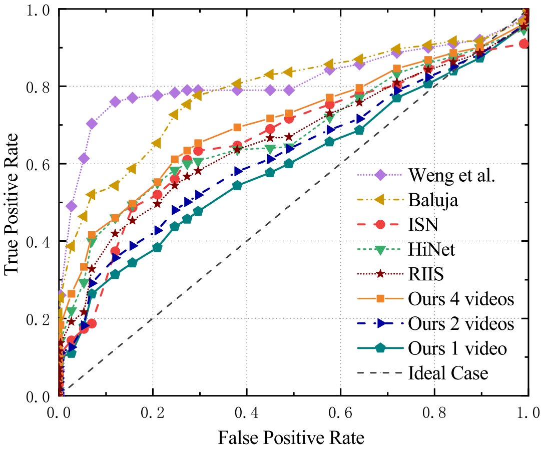


Figure 10. Statistics-based steganalysis by StegExpose [7]. The closer the detection accuracy is to 50%, the higher the security is.

steganalysis ability of different methods, which stands for the possibility of detecting stego frames from nature frames by steganalysis tools. We utilize the StegExpose [7] to attack different steganography methods. The detection set is built by mixing stego and cover with equal proportions. We vary the detection thresholds in a wide range in StegExpose and draw the receiver operating characteristic (ROC) curve in Fig. 10. Note that the ideal case represents that the detector has a 50% probability of detecting stego from an equally mixed cover and stego, the same as random sampling. Therefore, the closer the curve is to the ideal case, the higher the security is. Obviously, the stego frames generated by our LF-VSN are harder to be detected than other methods. Even in the multiple videos (*e.g.,* 2 and 4 videos) hiding, our method can still achieve attractive performance, demonstrating the higher data security of our LF-VSN.

4.6. Ablation Study

In this subsection, we present the ablation study in Tab. 4 to investigate the effect of different components in our LF-VSN. The experiments are conducted on Vimeo-T200.

Sliding window size. In this paper, we utilize the temporal correlation within each frame group to improve the video steganography performance. To demonstrate the effectiveness, we evaluate the performance of our LF-VSN with the window size $L = \{1, 3, 5\}$ in 2, 4, and 6 videos steganography. The results in Tab. 4 present that the temporal correlation has obvious performance gains to the multiple videos steganography. Considering the model complexity, we set the sliding window size as 3 in our LF-VSN.

Number of invertible blocks (IB). As mentioned above, our LF-VSN is composed of several IBs. To investigate the effectiveness of IB, we evaluate the performance of our LF-VSN with the number of IB being 12, 16, and 20. The results in Tab. 4 present that the performance increases with the number of IB. To make a trade-off between performance and complexity, we utilize 16 IBs in our LF-VSN.

Frequency concatenation (FreqCat). In our LF-VSN, we use the DWT transform to merge each input group in the frequency domain. To demonstrate the effectiveness, we replace this operation with direct channel-wise concatenation. Tab. 4 presents that there are $1.7dB$ and $1.91dB$ gains of FreqCat on stego and secret quality in 3 videos steganography. The possible reason is that DWT transform can separate the low-frequency and high-frequency sub-bands, making it more effective for information fusion and hiding.

Redundancy prediction module (RPM). In our LF-VSN, we employ RPM to predict the redundancy in the backward process instead of randomly sampling. To demonstrate the effectiveness of RPM, we replace this module with a random Gaussian sampling. The result in Tab. 4 shows that RPM can be used not only to design key-controllable steganography, but also to improve performance.

5. Conclusion

In this paper, we propose a large-capacity and flexible video steganography network (LF-VSN). The novelty of our method is twofold. First, our LF-VSN has a large hiding capacity, with which we can hide 7 **secret videos** into a cover video and then recover them well ($> 35dB$). Second, we explore the flexibility in multiple videos steganography by proposing a key-controllable scheme and a scalable design. Specifically, our key-controllable scheme can enable different receivers to recover particular secret videos through specific keys. Also, the key controlling is sensitive and model-specific, which can enhance data security. Our scalable design further improves the flexibility to hide a variable number of secret videos into a cover video with a single model. Extensive experiments demonstrate that our proposed LF-VSN has state-of-the-art performance with high security, large hiding capacity, and flexibility.

References

[1] Mahdi Ahmadi, Alireza Norouzi, Nader Karimi, Shadrokh Samavi, and Ali Emami. ReDMark: Framework for residual diffusion watermarking based on deep networks. *Expert Systems with Applications*, 146:113157, 2020. 3

[2] R Balaji and Garewal Naveen. Secure data transmission using video steganography. In *2011 IEEE International Conference on Electro/Information Technology*, pages 1–5, 2011. 2

[3] Shumeet Baluja. Hiding images in plain sight: Deep steganography. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 2069–2079, 2017. 1, 3

[4] Shumeet Baluja. Hiding images within images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(7):1685–1697, 2019. 1, 3, 6

[5] Mauro Barni, Franco Bartolini, and Alessandro Piva. Improved wavelet-based watermarking through pixel-wise masking. *IEEE Transactions on Image Processing*, 10(5):783–791, 2001. 1

[6] Debnath Bhattacharyya, Arup Kumar Bhaumik, Minkyu Choi, and Tai-hoon Kim. Directed graph pattern synthesis in lsb technique on video steganography. In *Advances in Computer Science and Information Technology*, pages 61–69. 2010. 1, 2

[7] Benedikt Boehm. Stegexpose-a tool for detecting lsb steganography. *arXiv preprint arXiv:1410.6656*, 2014. 8

[8] Chi-Kwong Chan and Lee-Ming Cheng. Hiding data in images by simple lsb substitution. *Pattern Recognition*, 37(3):469–474, 2004. 1

[9] Chantana Chantrapornchai, Kornkanok Churin, Jitdamrong Preechasuk, and Suchitra Adulkasem. Video steganography for hiding image with wavelet coefficients. *International Journal of Multimedia and Ubiquitous Engineering*, 9(6):385–396, 2014. 1, 2, 3

[10] Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. Digital image steganography: Survey and analysis of current methods. *Signal Processing*, 90(3):727–752, 2010. 1, 2

[11] Haoyu Chen, Linqi Song, Zhenxing Qian, Xinpeng Zhang, and Kede Ma. Hiding images in deep probabilistic models. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 6, 7

[12] Hans Delfs and Helmut Knebl. *Introduction to cryptography*, volume 2. Springer. 1

[13] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 1, 3

[14] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 1, 3

[15] Jessica Fridrich, Miroslav Goljan, and Rui Du. Detecting lsb steganography in color, and gray-scale images. *IEEE Transactions on Multimedia*, 8(4):22–28, 2001. 1, 3

[16] Jamie Hayes and George Danezis. Generating steganographic images via adversarial training. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 1951–1960, 2017. 3

[17] Chiou-Ting Hsu and Ja-Ling Wu. Hidden digital watermarks in images. *IEEE Transactions on Image Processing*, 8(1):58–68, 1999. 1

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017. 4

[19] Yan-Cheng Huang, Yi-Hsin Chen, Cheng-You Lu, Hui-Po Wang, Wen-Hsiao Peng, and Ching-Chun Huang. Video rescaling networks with joint optimization strategies for downscaling and upscaling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3527–3536, 2021. 3

[20] Tarik Faraj Idbeaa, Salina Abdul Samad, and Hafizah Husain. An adaptive compressed video steganography based on pixel-value differencing schemes. In *2015 International conference on advanced technologies for communications (ATC)*, pages 50–55, 2015. 2

[21] Junpeng Jing, Xin Deng, Mai Xu, Jianyi Wang, and Zhenyu Guan. HiNet: Deep image hiding by invertible network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4733–4742, 2021. 1, 3, 5, 6

[22] Richa Khare, Rachana Mishra, and Indrabhan Arya. Video steganography using lsb technique by neural network. In *2014 International Conference on Computational Intelligence and Communication Networks*, pages 898–902. IEEE, 2014. 1

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 6

[24] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 10236–10245, 2018. 3, 5

[25] Varsha Kishore, Xiangyu Chen, Yan Wang, Boyi Li, and Kilian Q Weinberger. Fixed neural network steganography: Train the images, not the network. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 3

[26] Kenneth C Laudon and Carol Guercio Traver. *E-commerce*. Pearson Boston, MA, 2013. 1

[27] Edward A Lee and David G Messerschmitt. *Digital communication*. Springer Science & Business Media, 2012. 1

[28] Daniel Lerch-Hostalot and David Megías. Unsupervised steganalysis based on artificial training sets. *Engineering Applications of Artificial Intelligence*, 50:45–59, 2016. 1, 3

[29] Jingyun Liang, Kai Zhang, Shuhang Gu, Luc Van Gool, and Radu Timofte. Flow-based kernel prior with application to blind super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10601–10610, 2021. 3

[30] Yang Liu, Zhenyue Qin, Saeed Anwar, Pan Ji, Dongwoo Kim, Sabrina Caldwell, and Tom Gedeon. Invertible denoising network: A light solution for real noise removal. In *Pro-*

ceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 13365–13374, 2021. 3

[31] Alessandro Lizzeri. Information revelation and certification intermediaries. *The RAND Journal of Economics*, 30(2):214–231, 1999. 1

[32] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10816–10825, 2021. 1, 3, 5, 6, 7

[33] Weiqi Luo, Fangjun Huang, and Jiwu Huang. Edge adaptive image steganography based on lsb matching revisited. *IEEE Transactions on Information Forensics and Security*, 5(2):201–214, 2010. 1, 3

[34] Jarno Mielikainen. Lsb matching revisited. *IEEE signal processing letters*, 13(5):285–287, 2006. 1, 2

[35] 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, 3

[36] Ian E Novos and Michael Waldman. The effects of increased copyright protection: An analytic approach. *Journal of Political Economy*, 92(2):236–246, 1984. 1

[37] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016. 7

[38] Mritha Ramalingam, Nor Ashidi Mat Isa, and R Puviarasi. A secured data hiding using affine transformation in video steganography. *Procedia Computer Science*, 171:1147–1156, 2020. 3

[39] JJKO Ruanaidh, WJ Dowling, and Francis M Boland. Phase watermarking of digital images. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 239–242, 1996. 1

[40] AP Sherly, PP Amritha, et al. A compressed video steganography using tpvd. *International Journal of Database Management Systems (IJDMS)*, 2(3):67–80, 2010. 2

[41] 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

[42] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2126, 2020. 3

[43] Xinyu Weng, Yongzhi Li, Lu Chi, and Yadong Mu. High-capacity convolutional video steganography with temporal residual modeling. In *Proceedings of the International Conference on Multimedia Retrieval (ICMR)*, pages 87–95, 2019. 1, 3, 6

[44] KokSheik Wong, Kiyoshi Tanaka, Koichi Takagi, and Yasuyuki Nakajima. Complete video quality-preserving data hiding. *IEEE Transactions on circuits and systems for video technology*, 19(10):1499–1512, 2009. 1, 2

[45] H-C Wu, N-I Wu, C-S Tsai, and M-S Hwang. Image steganographic scheme based on pixel-value differencing and lsb replacement methods. *IEE Proceedings-Vision, Image and Signal Processing*, 152(5):611–615, 2005. 1

[46] Mingqing Xiao, Shuxin Zheng, Chang Liu, Yaolong Wang, Di He, Guolin Ke, Jiang Bian, Zhouchen Lin, and Tie-Yan Liu. Invertible image rescaling. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 126–144, 2020. 3, 5

[47] Youmin Xu, Chong Mou, Yujie Hu, Jingfen Xie, and Jian Zhang. Robust invertible image steganography. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7875–7884, 2022. 6, 7

[48] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. 6

[49] Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust invisible video watermarking with attention. *arXiv e-prints*, pages arXiv–1909, 2019. 3

[50] Chen Zhao, Shuming Liu, Karttikeya Mangalam, and Bernard Ghanem. Re2tal: Rewiring pretrained video backbones for reversible temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3

[51] Rui Zhao, Tianshan Liu, Jun Xiao, Daniel PK Lun, and Kin-Man Lam. Invertible image decolorization. *IEEE Transactions on Image Processing*, 30:6081–6095, 2021. 3

[52] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 657–672, 2018. 3