

Generative Steganography via Live Comments on Streaming Video Frames

Yuling Liu , Cuilin Wang , Jie Wang , *Member, IEEE*, Bo Ou , and Xin Liao , *Senior Member, IEEE*

Abstract—Generative text steganography has received considerable attention in the covert communication community for the benefit of sending secret messages without the need to modify carriers. Existing methods typically choose the next word when generating a stego-text based on conditional probability encoding of candidates, which may lead to generating inadequate words for the underlying secret message. How to generate a semantically controllable stego-text with a high capacity on secure embedding of a secret message is a main challenge. We address this challenge by proposing a new paradigm to generative text steganography that takes advantage of certain social media through apparently normal behaviors from the sender. In particular, we make use of the live commenting feature provided by public video sharing platforms (PVSPs), which allow viewers to make comments on video scenes that will fly on screens when the scenes are shown. We show that this feature can be used to construct a generative steganographic system. The sender generates at random a number of distracting words and a certain invertible matrix called W - d matrix based on the total number of message words and distracting words. The sender then transforms a sequence of indexes of these words to a sequence, selects one or more videos with a sufficiently large number of total frames, and generates a comment on each frame in the sequence. The receiver extracts commented frame indexes, uses the shared stego-key to generate the same W - d matrix as the sender, and obtains the secret message using the inverse of the matrix. The stego-key consists of a vocabulary generator and a W - d matrix generator (WMG) based on pseudorandomly generated numbers. To generate comments on frames that conform to comments made by viewers, we devise a neural ResNet-LSTM model to generate a comment for an input image based on its content. Theoretical analysis shows that commented video frames (CVF) is covert, secure, efficient, and feasible to conceal any message of arbitrary length. We implement CVF and present evaluation results from multiple aspects that our work outperforms the existing stego-methods.

Index Terms—Generative steganography, live commenting on streaming videos, random W - d matrix, ResNet-LSTM network.

I. INTRODUCTION

DIGITAL steganographic methods (stego-methods, in short) use texts [1], [2], images [3], [4], videos [5], [6], and other digital forms to conceal secret messages for transmission over public channels. A digital object used to conceal secret messages is referred to a stego-carrier, and a digital object that contains a concealed message is referred to as a stego-object. A desirable stego-method should be covert, secure, efficient, and feasible. We refer to these requirements as CSEF requirements.

- 1) A stego-method is covert if it produces stego-objects indistinguishable from normal objects.
- 2) A stego-method is secure if it is computationally intractable for any adversary to extract from a stego-object the concealed message or partial information of it.
- 3) A stego-method is efficient if the following three conditions hold. a) It is computationally tractable for the sender to embed any secret message in the chosen carrier and for the receiver to extract the secret message from a stego-object. b) It is straightforward to generate stego-keys. c) It is straightforward for the sender to send and the receiver to receive stego-objects over a public channel without direct point-to-point communications.
- 4) A stego-method is feasible if it can conceal any secret message of arbitrary length.

Currently, the proposed stego-methods can be divided into three categories: modifying the stego-carrier methods, selecting the stego-object methods, and generating the stego-object methods [7]. Among them, the modification-carrier methods need to modify a given carrier to embed secret messages, so that they tend to be vulnerable to statistical attacks due to the modification traces [8], [9]. Once detected, the behavior of covert and secure communication fails. The selection-carrier methods help to resist such attacks [10], [11], but such methods tend to have limited embedding capacity and would need certain out-of-band information to select a stego-object, making it difficult to conceal longer messages. Therefore, it is difficult to meet the efficient-requirement and the feasible-requirement. Generative stego-methods can directly synthesize stego-objects or generate stego-objects to conceal secret messages without the need to modify carriers, which has attracted much attention in recent years. Since the image and text are the most widely used

Manuscript received 12 January 2023; revised 25 November 2023; accepted 4 January 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 61872134, Grant U22A2030, and Grant 61972142; in part by the National Key R&D Program of China under Grant 2022YFB3103500; in part by the Fujian Provincial Natural Science Foundation under Grant 2023J01246; and in part by the Key R&D projects in Hunan Province under Grant 2022SK2106. (*Corresponding authors: Yuling Liu; Jie Wang.*)

Yuling Liu, Bo Ou, and Xin Liao are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: yuling_liu@hnu.edu.cn; oubo@hnu.edu.cn; xinliao@hnu.edu.cn).

Cuilin Wang is with Xiangxi Power Supply Branch, State Grid Hunan Electric Power Company, Ltd., Jishou 416000, China (e-mail: cuilinwang@hnu.edu.cn).

Jie Wang is with the Richard Miner School of Computer & Information Sciences, University of Massachusetts, Lowell, MA 01854 USA (e-mail: wang@cs.uml.edu).

Digital Object Identifier 10.1109/TCSS.2024.3352979

carriers, more researchers focus on the field of generative image steganography and generative text steganography.

Early generative image stego-methods typically conceal secret messages in certain kinds of images of desired particularities, such as image texture [12] and fingerprint [13]. However, texture and fingerprint images are not normal images. Generative adversarial networks (GANs), a recently developed neural-net model, have been used to generate stego-images that look normal, for GAN is able to generate natural-looking images. For example, Liu et al. [14] devised a method to encode secret messages into the corresponding category labels and use a labeled auxiliary classifier GAN neural-net model to generate stego-images. Hu et al. [15] presented a method to convert a secret message into a low-dimensional vector of noise signal as the input of a deep convolutional GAN model for generating a stego-image. Stego-images generated by GAN-based models, unfortunately, often suffer from low visual-quality with limited hiding capacity. These images may arouse the attacker's suspicion that this is an stego-image, and so fail the covert-requirement.

Different from generative image stego-methods, generative text steganography can be divided into two types: semantically uncontrollable and semantically controllable. Semantically uncontrollable stego-methods generate text freely. Early methods suffer from problems of incoherent semantics or incorrect grammar [16], [17]. Syntactic rules are used to guide the generation of text. Semantically uncontrollable stego-methods based on deep neural models have been proposed successively [18], [19], [20], [21], which generate texts that are semantic coherence and grammatical. However, the generated texts do not conform to texts written by educated authors with the following problems: Content semantics may look random without themes, text may be repeated, irrelevant, or uninformative. Semantically controllable stego-methods intend to generate texts on certain topics or meet the specification of control information [22], [23], [24].

Since generative methods open a new horizon to construct steganographic systems [25], [26], [27], [28]. In addition to standard carriers of texts and images, online social-network applications such as blogs and public video sharing platforms (PVSPs) provide a rich source for exploring new generative stego-methods. For example, the live-commenting feature on PVSP allows viewers to post text comments on any frame of a video, which fly across the video showing screen when it is shown the next time, visible to all viewers [29]. We devise a new generative stego-method called commented video frames (CVF) that combines texts and videos through commented videos posted on a PVSP as a stego-carrier.

In particular, the sender represents a secret message with a number of randomly generated distracting words as a sequence of word indexes in a vocabulary and transform this sequence using a frame-generation algorithm to a sequence of strictly increasing indexes of video frames with a desired spread. The sender then selects one or more videos with a sufficiently large number of frames, uses a neural model to generate comments according to the contents of the frames in the sequence, and posts them in the order of the sequence of frames to form a stego-object. The receiver, upon receiving the commented

videos posted by the sender, runs the message-extraction algorithm on the commented video(s) to extract the secret message based on the sequence of indexes of the commented frames. Both the frame-generation algorithm and the message-extraction algorithm use the same stego-key, which consists of a randomly permuted vocabulary sufficient for writing secret messages, and a generator of a special kind of matrices called W - d matrices that are generated pseudorandomly. We argue that CVF meets the CSEF requirements. In this work, our contributions are as follows.

- 1) We propose a new generative stego-method that combines texts and videos through commented videos posted on a PVSP as a stego-carrier, which leverages the live commenting feature on videos to achieve steganography.
- 2) Randomly generated distracting words are represented a secret message, and frame-sequence-generator (FSG) is introduced to select the frame sequences for embedding. Theoretical analysis shows that the proposed method can be covert, secure, efficient, and feasible, and conceal secret messages of arbitrary lengths in one or more videos.
- 3) We evaluate the proposed method via experiments from multiple aspects, and the experimental results demonstrate the superiority of our work over the current state-of-the-art stego-methods.

The rest of the article is organized as follows. Section II is the architecture and data flow of CVF. We explain how to generate comments on given frames in Section III and argue that CVF meets the CSEF requirements in Section IV. We implement CVF and present evaluation results in Section V and conclude the article in Section VI.

II. CVF ARCHITECTURE AND DATA FLOW

CVF consists of the following five components: 1) stego-key; 2) distracting-word generator (DWG); 3) frame-sequence generator (FSG); 4) frame-comment generator (FCG); and 5) secret-message extractor (SME), where the stego-key consists of a vocab generator (VoG) and a W - d matrix generator (WMG). The stego-key and the underlying parameters are shared between the sender and the receiver via a secret channel before sending and receiving stego-objects on a PVSP and must be kept secret. Fig. 1 depicts the architecture of CVF and its data flow.

A. Stego-Key Generation

1) *VoG*: Let V_0 be a vocabulary of commonly used words, digits, and punctuation sufficient to write secret messages. Let \mathcal{P} be a fixed pseudorandom permutation algorithm to permute integers from 1 to $|V_0|$ based on a fixed seed function $s_0(n)$ that depends on n (e.g., $s_0(n) = (a_0 + b_0n)^2$ with positive integers a_0 and b_0), where n is an integer depending on the secret message (e.g., we may let n be the length of the message). VoG generates V by permuting V_0 using \mathcal{P} so that words in V are listed randomly, rather than lexicographically, where V consists of a word-to-index table and an index-to-word table for fast retrieval, with index starting from 1.

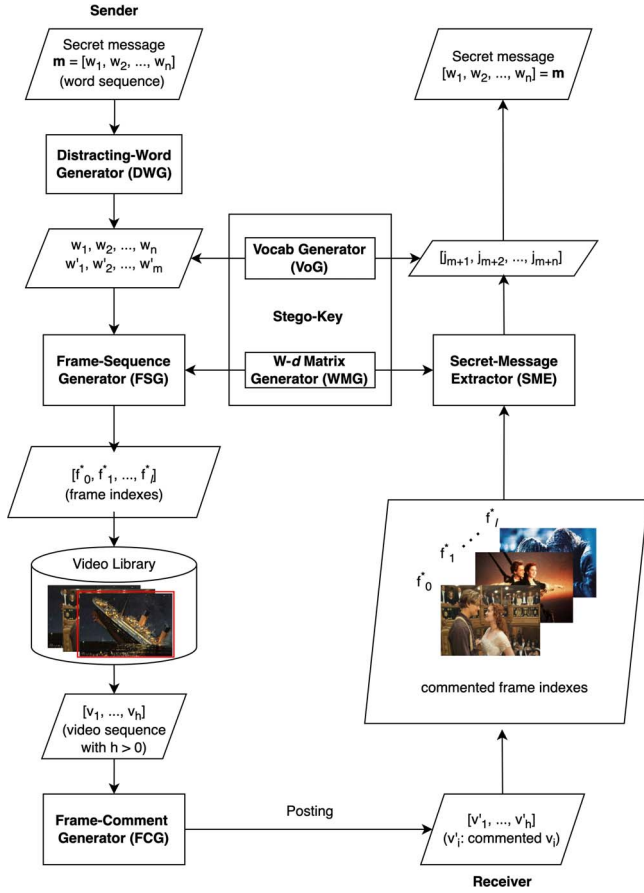


Fig. 1. CVF architecture and data flow.

2) **WMG**: WMG uses a fixed pseudorandom bit generator (PRBG) \mathcal{B} , a fixed pseudorandom number generator (PRNG) $\mathcal{N}(a, b)$ that generates a positive integer between a and b with integers $a < b$, and a fixed seed function $s_1(\ell)$ depending on ℓ [e.g., $s_1(\ell) = (a_1 + b_1\ell)^2$ with positive integers a_1 and b_1].

Given an integer ℓ , WMG generates an $\ell \times \ell$ W-d matrix $M = [m_{ij}]_{\ell \times \ell}$ as follows, using d as the initial value for its diagonal.

- 1) Run \mathcal{B} with seed $s(\ell)$ to generate $\ell - 1$ values $d_1, d_2, \dots, d_{\ell-1}$ with $d_j \in \{0, 1\}$.
- 2) Let M_0 be an $\ell \times \ell$ matrix such that each entry in the diagonal is d , and the first row is $[d, d_1, \dots, d_{\ell-1}]$.
- 3) Generate M from M_0 as follows: (1) The first row in M is the same as the first row in M_0 . (2) The $(i + 1)$ -th row in M is the summation of the i th row in M and the $(i + 1)$ -th row in M_0 , for $i = 1, \dots, \ell - 1$.

The following is an example a 5×5 W-d matrix, where d^+ represents $d + 1$:

$$M_0 = \begin{bmatrix} d & 0 & 1 & 1 & 0 \\ 0 & d & 0 & 0 & 0 \\ 0 & 0 & d & 0 & 0 \\ 0 & 0 & 0 & d & 0 \\ 0 & 0 & 0 & 0 & d \end{bmatrix}, \quad M = \begin{bmatrix} d & 0 & 1 & 1 & 0 \\ d & d & 1 & 1 & 0 \\ d & d & d^+ & 1 & 0 \\ d & d & d^+ & d^+ & 0 \\ d & d & d^+ & d^+ & d \end{bmatrix}.$$

Property 1: M is invertible.

Property 2: For any sequence of ℓ positive integers $x = [x_1, x_2, \dots, x_\ell]$, treated as an ℓ -dimensional vector, we have that $M \cdot x$ is a sequence of ℓ strictly increasing positive integers $y = [y_1, y_2, \dots, y_\ell]$ with $y_{i+1} - y_i \geq d$ for $1 \leq i < \ell$.

Property 3: M is a matrix with entries in $\{0, 1, d, d + 1\}$ with the following properties. 1) Every entry in the first column is d . 2) If the j th entry in the first row of M_0 is 0, then for the j th column of M , we have $m_{ij} = 0$ for $1 \leq i < j$ and $m_{ij} = d$ for $j \leq i \leq \ell$. 3) If the j th entry in the first row of M_0 is 1, then for the j th column of M , we have $m_{ij} = 1$ for $1 \leq i < j$ and $m_{ij} = d + 1$ for $j \leq i \leq \ell$.

To see why Property 1 is true, it suffices to note that M is obtained from elementary row operations on M_0 and M_0 has a full rank because its determinant is equal to $d^\ell \neq 0$. For example, let M be the W-d matrix in the above example, then

$$M^{-1} = \begin{bmatrix} d^{-1} & d^{-2} & 0 & -d^{-2} & 0 \\ -d^{-1} & d^{-1} & 0 & 0 & 0 \\ 0 & -d^{-1} & d^{-1} & 0 & 0 \\ 0 & 0 & -d^{-1} & d^{-1} & 0 \\ 0 & 0 & 0 & -d^{-1} & d^{-1} \end{bmatrix}.$$

To see why Property 2 is true, it suffices to note that $y_{j+1} = r_i \cdot x + d \cdot x_{j+1}$, where r_i is the i th row in M , $r_i \cdot x$ is the inner product equal to y_i , and $x_{j+1} \geq 1$.

To see why Property 3 is true, it suffices to note that each column in M_0 contains exactly one d in the diagonal, at most one in the first row, and zero in the remaining entries.

Both the sender and the receiver share the same stego-key generation algorithm, and they must agree beforehand through a separate channel on the following parameters: V_0 , \mathcal{P} , $s_0(\cdot)$, \mathcal{B} , $\mathcal{N}(a, b)$, $s_1(\cdot)$, and d . Both parties can generate the same V and W-d matrix $[m_{ij}]_{\ell \times \ell}$ with a given ℓ . This makes stego-key generation a straightforward process.

Remark: The above properties of W-d matrices are true as long as $d \neq 0$.

B. DWG

A secret message m is a sequence of words in vocabulary V_0 , written as

$$m = [w_1, w_2, \dots, w_n].$$

DWG first generates pseudorandomly a positive integer m and then generates m distracting words in V_0 .

A word w is considered distracting for m if knowing w does not reveal any information m contains. This may be done, for example, as follows. Let

$$S_m = \bigcup_{i=1}^n S_{w_i}$$

where S_w is the set of synsets of w in WordNet [30]. If $S_w \cap S_m = \emptyset$, then w is a distracting word for m . We may find distracting words efficiently as follows. Let w be a message word. Start from w on the WordNet tree, traverse up two or more levels to a hypernym w_h , traverse down from w_h on a different branch two more levels, and use a hyponym w' at that level as a distracting word of w . Let $\{w'_1, w'_2, \dots, w'_m\}$ be a set of distracting words being selected for m .

C. FSG

Generate V from V_0 using \mathcal{P} with seed $s_0(n+m)$. Let j_k and j_{m+i} be, respectively, the index for distracting word w'_k and message word w_i in V , with $1 \leq k \leq m$ and $1 \leq i \leq n$. Denote n by j_{m+n+1} . Let

$$j_{\max} = \lfloor \log(\max\{j_i \mid i = 1, \dots, m+n+1\}) \rfloor$$

and $u = j_{\max} + 1$. For each j_i , use the fast exponentiation algorithm to compute $b_{i,0}, \dots, b_{i,u}$, where $b_{i,j} \in \{1, 2\}$, such that

$$\sum_{j=0}^{u-1} (b_{i,j} - 1)2^j = j_i.$$

Represent j_i by $[b_{i,0}, \dots, b_{i,u-1}]$. Let

$$\mathbf{J} = [b_{1,0}, \dots, b_{1,u-1}, \dots, b_{m+n+1,0}, \dots, b_{m+n+1,u-1}, u].$$

Let $\ell = |\mathbf{J}| = u(m+n+1) + 1$. FSG first generates a W - d matrix $\mathbf{M}_{\ell \times \ell}$ using the procedure described in Section II-A2. It then computes

$$\mathbf{F} = \mathbf{M} \cdot \mathbf{J} = [f_1, f_2, \dots, f_\ell].$$

The following properties are straightforward.

Property 4: $f_1 < f_2 < \dots < f_\ell$ and $f_i + d \leq f_{i+1}$ for $i = 1, 2, \dots, \ell - 1$.

Property 5: $u \leq \lfloor \log |V| \rfloor + 1$.

Property 6: $f_\ell < (d+1)(2\ell + u - 2)$.

Property 4 is a corollary of Property 2.

To see why Property 5 is true, it suffices to note that $j_i \leq |V|$ for all i .

To see why Property 6 is true, it suffices to note from Property 3 that each entry in the W - d matrix \mathbf{M} is in $\{0, 1, d, d+1\}$ and so the largest possible value that is generated by $\mathbf{M} \cdot \mathbf{J}$ is when each entry in the first row of \mathbf{M}_0 , except m_{11} , is 1, and all the entries in \mathbf{J} , except u , is 2. When this happens, the first row of \mathbf{M} is $[d, 1, \dots, 1]$ and for the i th row, $m_{i1} = d$, $m_{ij} = d+1$ for $2 \leq j \leq i$, and $m_{ij} = 1$ for $i < j \leq \ell$. The last row of \mathbf{M} is $[d, d+1, \dots, d+1]$. Thus,

$$\begin{aligned} f_1 &= 2(d + \ell - 2 + u) \\ f_i &= 2d + 2(i-1)(d+1) + 2(\ell - i - 1) + u, \\ &\quad \text{for } i = 2, \dots, \ell - 1 \\ f_\ell &= 2d + (d+1)(2(\ell-2) + u) < (d+1)(2\ell + u - 2). \end{aligned}$$

Next, WMG generates a random number r using $\mathcal{N}(a_2, b_2)$ with seed $s(\ell)$. Let

$$\begin{aligned} \lambda &= \lfloor \log f_1 \rfloor \\ f_0^* &= r + \lambda \\ f_i^* &= f_0^* + f_i - 2^\lambda + d \quad (i = 1, \dots, \ell). \end{aligned}$$

FSG outputs

$$\mathbf{F}^* = [f_0^*, f_1^*, \dots, f_\ell^*].$$

Property 7: $f_{i-1}^* + d \leq f_i^*$ for $i = 1, \dots, \ell$.

To see why Property 7 is true, it suffices to note that $f_1^* - f_0^* \geq d$, and for $i \geq 1$, we have $f_{i+1}^* - f_i^* = f_{i+1} - f_i$, which is at least d from Property 4.

Remark: (1) For practical applications, V may be generated and fixed before sending and receiving concealed messages. Generating V for each secret message may add to the security with a small cost of permuting V_0 . (2) \mathbf{F}^* is essentially a shift of \mathbf{F} to the left by $2^{\lfloor \log f_1 \rfloor}$ positions. It is possible to just use \mathbf{F} as output. Using \mathbf{F}^* as output increases embedding capacity—see Section V-G for an analysis.

D. Video Selection

CVF chooses one or more videos from the video library provided by the underlying PVSP such that the total number of frames is greater than f_ℓ^* . Let v_1, \dots, v_h with $h > 0$ be the selected videos with F_{v_i} being the number of frames of v_i such that $\sum_{k=1}^h F_{v_k} > f_\ell^*$ and $\sum_{k=1}^{h-1} F_{v_k} < f_\ell^*$. Index the j th frame of v_i as $\sum_{k=1}^{i-1} F_{v_k} + j$ to index the frames in the selected videos from 1 to $\sum_{k=1}^h F_{v_k}$.

Remark: In practical applications, it may be desirable to spread one message across multiple videos, even if the message can be concealed in one video. This can be done as follows. 1) Split \mathbf{F}^* into h segments $\mathbf{F}_1^*, \dots, \mathbf{F}_h^*$ of various sizes with $h > 1$ chosen by the need of spreading. 2) Let $n_j = |\mathbf{F}_j^*|$ and f_{j,n_j}^* be the last element in \mathbf{F}_j^* . For $j > 1$ let $f_{j,\kappa}' = f_{j,\kappa}^* - \sum_{i=1}^{j-1} f_{i,n_i}^*$ for $\kappa = 1, \dots, n_j$. 3) Let $\mathbf{F}_1' = \mathbf{F}_1^*$ and $\mathbf{F}_j' = [f_{j,1}', \dots, f_{j,n_j}']$ for $1 < j \leq h$. 4) Select a sequence of h videos v_1, \dots, v_h such that \mathbf{F}_j' can be concealed in v_j and generate commented video v_j' accordingly. It is straightforward to reverse the index of commented frames in v_1', \dots, v_h' back to \mathbf{F}^* .

E. FCG

Suppose that f_i^* falls in video v_p as the q th frame. FCG is a ResNet-LSTM neural model that generates a comment on a given frame based on the content of the frame image and posts it on the frame. FCG then posts the commented videos v_1', \dots, v_h' in this order on PVSP. Detail description of FCG is presented in Section III.

F. SME

SME on the receiver side first extracts from the commented video(s) v_1', \dots, v_h' the sequence of frames

$$\mathbf{F}^* = [f_0^*, f_1^*, \dots, f_\ell^*].$$

In particular, let j be a commented frame index in v_i . Then, $f_i^* = \sum_{k=0}^{i-1} F_{v_k} + j$.

SME then computes the same r as FSG using $\mathcal{N}(a, b)$ with seed $s_1(\ell)$ to compute $\lambda = f_0^* - r$, and then, f_i one by one for $i = 1, \dots, \ell$ to get \mathbf{F} .

Next, SME generates the same W - d matrix \mathbf{M} as FSG and computes $\mathbf{J} = \mathbf{M}^{-1} \mathbf{F}$ to get \mathbf{J} , from which it can get u, n, m , and $[j_{m+1}, j_{m+2}, \dots, j_{m+n}]$. Finally, generate V using \mathcal{P}

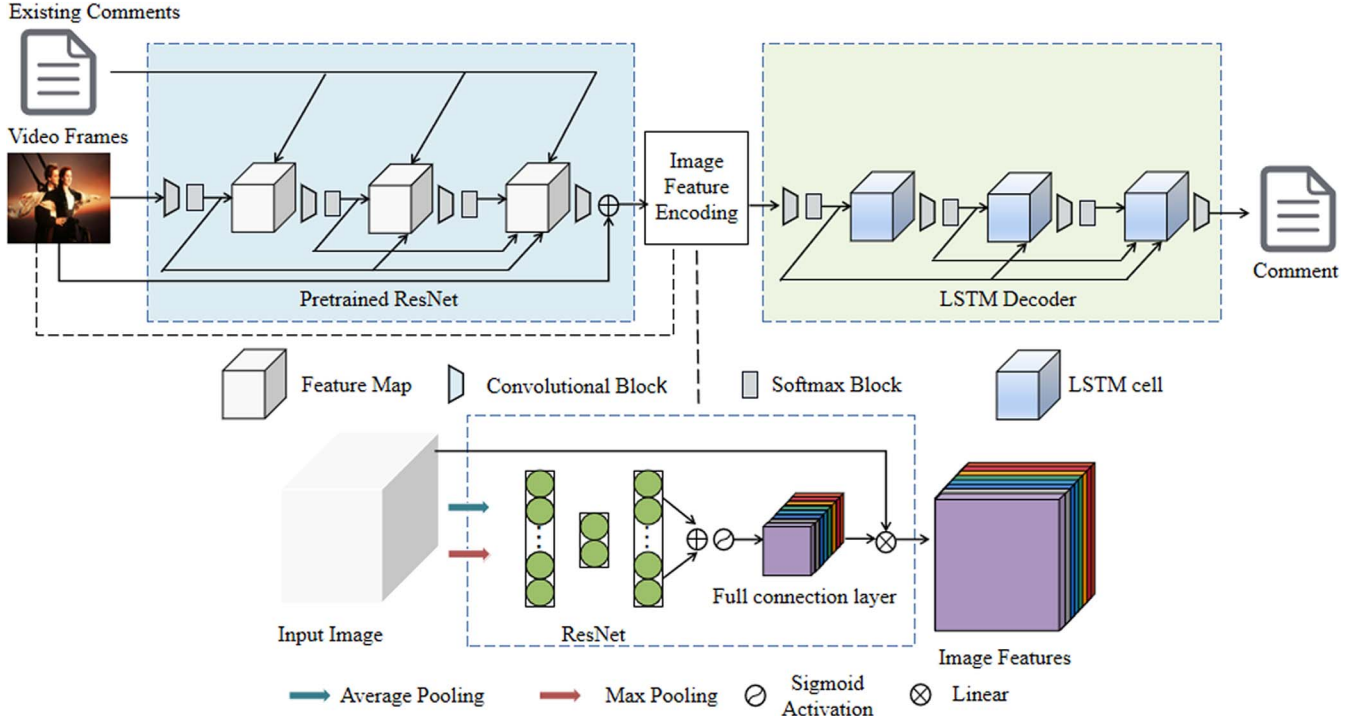


Fig. 2. FCG structure: a ResNet encoder and an LSTM decoder.

with seed $s_0(m+n)$ and use the vocabulary V to obtain the original message

$$[w_1, w_2, \dots, w_n] = \mathbf{m}.$$

Note that entries of M^{-1} may not be integers, and so the i th entry in $M^{-1}\mathbf{F}$ may incur a small roundoff error, which, when rounding to integer, returns the same i th entry in \mathbf{J} . In other words, the roundoff errors are too small to cause any negative effect. This is confirmed by extensive numerical experiments (see Section V-G for details).

III. DESCRIPTION OF FCG

FCG is a neural model to extract features from a frame image using a residual network (ResNet) encoder and generates a short comment on the image using an long short-term memory (LSTM) decoder. Fig. 2 depicts the structure of FCG.

A. ResNet Encoder

We modify an existing ResNet model to extract features from images using shortcut connections with the following framework:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{\mathbf{W}_i\}) + \mathbf{W}_s \cdot \mathbf{x}$$

where \mathbf{x} and \mathbf{y} are, respectively, the input and output vectors of a sequence of multiple layers $\{\mathbf{W}_i\}$ with \mathbf{W}_i being the weight matrix at layer i , the function $\mathcal{F}(\mathbf{x}, \{\mathbf{W}_i\})$ is the

residual mapping to be learned, and \mathbf{W}_s is a linear projection. For example, for a two-layer short cut on $\{\mathbf{W}_1, \mathbf{W}_2\}$ and a three-layer shortcut on $\{\mathbf{W}_3, \mathbf{W}_4, \mathbf{W}_5\}$

$$\mathcal{F}(\mathbf{x}, \{\mathbf{W}_1, \mathbf{W}_2\}) = \mathbf{W}_2 \cdot \sigma(\mathbf{W}_1 \cdot \mathbf{x}),$$

$$\mathcal{F}(\mathbf{x}, \{\mathbf{W}_3, \mathbf{W}_4, \mathbf{W}_5\}) = \mathbf{W}_5 \cdot \sigma(\mathbf{W}_4 \cdot \sigma(\mathbf{W}_3 \cdot \mathbf{x}))$$

where σ is an activation function (e.g., $\sigma = \text{ReLU}$).

We modify a pretrained ResNet model [31] by replacing its last layer with an identity mapping to generate a 256-dimensional vector \mathbf{I} as the input of the LSTM decoder.

B. LSTM Decoder

The task is to generate short yet semantically complete comments based on the image-feature vector \mathbf{I} generated by the ResNet encoder and (if any) the existing comments on the same frame $\mathbf{x} = [x_1, x_2, \dots, x_k]$. In particular, the decoder first converts the comment into a 256-dimensional vector through an embedded layer, splices it with \mathbf{I} , and then fits the output into an LSTM structure to generate comments.

1) *LSTM Structure*: Motivated by an existing model [18], we use an RNN structure to decode. Traditional RNN has no memory function and so cannot handle long texts. Hence, we devise a variant using a gated mechanism to remember selected information. Fig. 3 depicts the structure of this variant.

2) *Distribution Learning*: The LSTM component controls the transmission of information with three gates: the input gate, the forget gate, and the output gate. The following are the

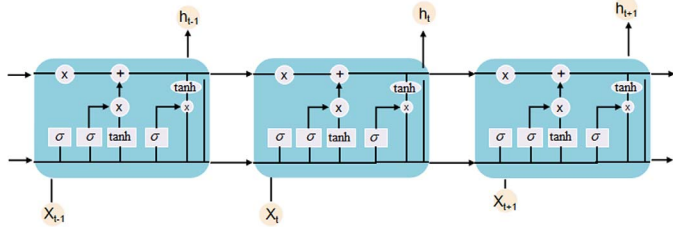


Fig. 3. Internal structure of the LSTM variant.

state update, the hidden layer transmission, and the final output parameter update:

$$\begin{cases} i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\ f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \\ o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t = o_t \cdot \tanh(C_t) \end{cases}$$

where i_t represents the input gate, f_t the forget gate, and o_t the output gate; W_i , W_c , W_f , and W_o are weight matrices; b_i , b_c , b_f , and b_o are bias; h_{t-1} represents the output of the previous unit, and x_t represents the current input.

In particular, the LSTM component first determines how much information is discarded in the last state C_{t-1} through f_t . Next, it uses a tanh layer to create a candidate vector \tilde{C}_t to add to the state of the network unit. It then updates the state according to the results calculated in the previous steps and computes its value as C_t . Finally, it determines which information of each unit is the output through a sigmoid function.

Denote the transfer function of LSTM units by $f_{\text{LSTM}}(\cdot)$. The LSTM model can learn the statistical language model from a large number of normal texts, calculate the probability distribution of the next word according to the previous generated words, and finally realize the generation of sentences. For example, assuming that the model has generated $i-1$ words and will generate the i th word, the model first calculates the probability distribution of the i th word according to the image feature vector I and the generated $i-1$ words. That is

$$c_{i-1} = f_{\text{LSTM}}(x_1, x_2, \dots, x_{i-1}, I)$$

where c_{i-1} represents the output vector of LSTM. We calculate the probability distribution of the i th word as follows: Write the current probability as a matrix $C_p \in \mathbb{R}^{r \times N}$ with $r = 256$

$$C_p = \begin{bmatrix} w_{1,1}^p & w_{1,2}^p & \cdots & w_{1,N}^p \\ w_{2,1}^p & w_{2,2}^p & \cdots & w_{2,N}^p \\ \vdots & \vdots & \ddots & \vdots \\ w_{r,1}^p & w_{r,2}^p & \cdots & w_{r,N}^p \end{bmatrix}$$

$$y_i = \sum_{k=1}^r w_{k,i}^p \cdot o_{i,t}^l + b_{i,t}^p.$$

C_p and b^p are the matrix and the bias of learned weight. We use matrix C_p calculated at the current time to obtain the score

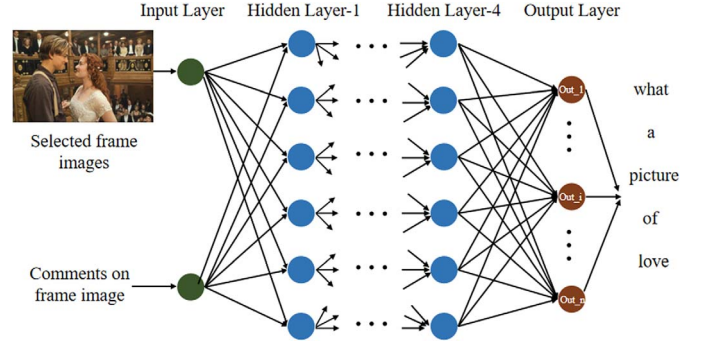


Fig. 4. Comment generation.

of each word y_i in the candidate words. We added a softmax classifier layer to calculate the distribution probability of the next word in each step

$$p(w_j | x_1, x_2, \dots, x_{t-1}, I) = \frac{\exp(y_i)}{\sum_{j=1}^N \exp(y_i)}$$

where w_j represents the selected word in the vocabulary V , and the dimension of the output vector y is the same as that of the input vector x .

3) *Distribution Selection*: To compute a probability distribution of word y_i , we use beam search to select words for generating a comment (see Section V for detail). Fig. 4 is an example of the comment generation process.

IV. CVF AND THE CSEF REQUIREMENTS

We argue that CVF meets the CSEF requirements. We refer to the posted commented video(s) produced by FCG as a CVF-object.

A. Covertness

The live commenting feature on videos provided by PVSPs allows viewers to pause a video at a chosen frame and enter comments. The comments are stored on the corresponding frame with time stamps when they are entered. These comments, after passing inspection of the underlying PVSP that they are adequate to post, are shown on the video-displaying screen the next time when the video is played. Viewers typically post comments following the playing order of the video, that is, on frames with time stamps in increasing order, whose underlying frame indexes are also in increasing order.

CVF transforms J into a sequence F^* of frame indexes in increasing order to be commented on, with a controllable gap between consecutive comments on frames to produce a sufficient spread. Moreover, there are no patterns on the size of F^* and the gaps between two consecutive frames in F^* . These conform to the norm of where a viewer would post comments. Thus, checking commented frames and time stamps on posted comments would not arouse suspicions of the adversary that the underlying commented video is a stego-object.

Next, comments generated by CVF are based on the contents of the underlying frames and the comments are short, diverse,

and both semantically and grammatically correct that conform to native speakers (see Section V-D for detail). Thus, checking posted comments would not arouse suspicions of the adversary.

B. Security

With the rapid development of various emerging technologies, such as telehealth systems, IoT-driven smart cities, and social media, people urgently need to protect the privacy of personal information. For many cases, blockchain has been a predominant technology meeting this demand. For example, Younis et al. [32] presented a novel solution that enables secure storage and dissemination of patient data and puts patients in charge of defining access rules. Bayar et al. [33] presented an intelligent blockchain framework that integrates blockchain with machine learning techniques to protect privacy and security in IoT-driven smart cities. Different from the works on blockchain based privacy-preserving, we devise a new generative stego-method called CVF that combines texts and videos through commented videos posted on a PVSP as a stego-carrier. An CVF-object does not reveal the length of the concealed secret message even within a reasonable range because of the m added distracting words with m being generated at random in a large range.

The sequence of frame indexes revealed by a CVF-object does not reveal word indexes or any structure because of the randomnesses of the W - d matrix M and the random permutation of the vocabulary.

While the attacker may brute-force all possible M , doing so would require to try $d^* \times 2^{\ell-1}$ different initial matrices M_0 and $N!$ permutations of V_0 , where d^* is the upper bound of d , N is the total number of words in V . This is intractable.

Suppose that the vocabulary V is known, but the W - d matrix M remains a secret. This is the same as above except that a brute-force attack only involves M_0 , which incurs $d^* \times 2^{\ell-1}$ trials. Thus, as long as ℓ is sufficiently large, for example, $\ell > 200$, brute-forcing is intractable.

Suppose that the W - d matrix M is known, but the vocabulary V remains secret. Although attackers may now obtain the sequence of word indexes, they still have no way to know which words correspond to these indexes because of the random permutation of V_0 . A brute-force attack would involve $N!$ trials, which is intractable.

C. Efficiency

The vocabulary V , which can be generated beforehand, is a random permutation of V_0 that can be carried out efficiently. Given a secret message m , the sender follows a straightforward construction to generate a W - d matrix M . Using it to transform J to F^* can be done efficiently.

Generating comments on frames in F^* can be done efficiently after the ResNet-LSTM model is trained. Its time complexity is negligible compared to the computing time of F^* .

Upon receiving a CVF-object through PVSP, it is straightforward for the receiver to extract F^* . The receiver then uses the same procedure to generate M . Computing M^{-1} can be done efficiently, from which it is straightforward to compute

m . Section V-G provides time complexity analysis of CVF via numerical experiments.

Making a new stego-key can be done efficiently by changing some or all of the parameters in the stego-key generation algorithm.

D. Feasibility

It is a general consensus¹ that knowing 1000 to 3000 words allows people to carry on everyday conversations. Knowing 4000 to 10 000 words makes people advanced language users while knowing more than 10 000 words puts them at the fluent or native-speaker levels. Without loss of generality, assume that $|V_0| \leq 2^{14} = 16\,384$, which is deemed sufficient to express any secret message between native speakers.

A video's rate of frames is at least 24 frames per second (FPS); some may have 60 or larger FPS. Without loss of generality, we assume that a 2-h video has at least 172 800 frames.

It follows from the proofs of Properties 4 and 6 that when f_ℓ reaches the maximum value, we have $f_{i+1} - f_i \geq 2d$ for $i \geq 1$, and so it follows from Property 7 that $f_{i+1}^* - f_i^* \geq 2d$. It follows from Property 6 that $f_\ell < (d+1)(2\ell+u-2)$ and so

$$f_\ell^* < (d+1)(2\ell+u-2) - 2^{\lambda^*} + r + \lambda^* + d$$

where $\lambda^* = \lfloor \log(2(d+\ell-2+u)) \rfloor$ and r is a number generated by PRNG $\mathcal{N}(a_2, b_2)$. Thus,

$$\begin{aligned} f_\ell^* &< (d+1)(2\ell+u-2) - \frac{2(d+\ell-2+u)}{2} + r + u + d \\ &= (2d+1)\ell + d(u-2) + u + r. \end{aligned}$$

It follows from Property 4 that $u \leq 15$. For the sake of demonstrating worst-case embedding capacity that preserves covert-ness, let $a = 1440$, $b = 2a$, and $d = 6$. This ensures that the first comment is made after at least one minute, and the next comment is at least half of a second after the current one. This implies that $f_\ell^* < 13\ell + 3059$. Note that $\ell = u(m+n+1) + 1 \leq 15(m+n+1) + 1$. Letting $13 \cdot 15(m+n+1) + 13 + 3058 \leq 172\,800$, we get $m+n+1 \leq 870$. This means that in the worst case a 2-h video can conceal at least 870 words, which is deemed sufficient for concealing most secret messages with adequate numbers of distracting words. We note that in most cases, a 2-h video can actually conceal substantially more words. For example, if we generate word indexes independently at random using a PRNG between 1 and 16 384 with $d = 6$, then a 2-h video can conceal over 2300 words (see Section V-G for experiment details).

A PVSP typically offers a collection of thousands of videos in its library. Thus, CVF can practically conceal messages of any length in one or more videos on any reasonable PVSP.

Remark: If we use F as output in FSG, then we have $f_\ell < (d+1)(2u(m+n+1) + 2 + u - 2) \leq 14 \cdot 15(m+n+2)$. Letting $14 \cdot 15(m+n+2) \leq 172\,800$, we get $m+n \leq 820$. In this case, F^* can embed 6% more words than F .

¹<https://www.optilingo.com/blog/general/how-many-words-do-you-need-to-know-to-become-fluent-in-a-language/>

TABLE I
TRAIN-VALIDATION-TEST SPLITS OF FLICKR8K AND BILI2CV

	Train	Validation	Test
Flickr8k	6000	1000	1000
Bili2CV	3000	1000	1000

V. EVALUATIONS

We evaluate CVF via experiments from multiple aspects and present evaluation results.

A. Datasets

Flickr8k [34] is a dataset publicly available, which consists of 8000 images with a factual descriptions on each image that describes objects contained in a given image and relationships between them. However, using Flickr8k alone to train a FCG/ResNet-LSTM model can only generate a factual description of a given image, which does not look like a comment made by a viewer. Comments made by viewers on images are colloquial, diverse, emotional, and sometimes abstract, which are much different from factual descriptions. To generate such comments, we would need a large number of commented frame images with comments made by different viewers. To the best of our knowledge, no such datasets are publicly available. Thus, we construct our own dataset by crawling commented videos from bilibili.com, a popular PVSP in mainland China, and we call the dataset Bili2CV (a short hand for Bilibili Commented Videos).

Bili2CV is constructed as follows: We segment each commented video according to its frame rate and pair each commented frame with its comments to obtain multiple references. We collect the most representative videos and use top ten pages of comments for each commented video. Bili2CV covers 15 categories: animation, food, game, motion, wear, skincare, cook, movie, vlog, tour, dance, car, music, knowledge, and technology. We remove duplicated short videos and filter videos with low quality or with a small number of comments to ensure data quality. We then manually translate comments written in Chinese to English so that we can use Flickr8k and Bili2CV together. Table I shows the statistics of Flickr8k and Bili2CV with train-validation-test splits.

Fig. 5 shows a sample comment generated by FCG trained on Flickr8k alone, and a sample comment generated by FCG trained on Flickr8k and Bili2CV together, from which we can see that the former looks like a description of the image while the latter looks more like a live comment made by a viewer.

B. Encoder

We compare a number of models on extracting image features, including ResNet 152 [31], VGG 16 [35], and Inception V3 [36], using a NVIDIA GeForce RTX 2080 Ti graphics card. All models are trained with 20 epochs and batch size of 64. The loss rate on the training set and the validation set is shown in Fig. 6, from which it is evident that ResNet 152



Comment: “A woman is sitting on a bench in front of a building.”

(a)



Comment: “Skin is so pale.”

(b)

Fig. 5. (a) Generated comment by FCG trained over Flickr8k, which is a description of the image. (b) Generated comment by FCG trained over Flickr8k \cup Bili2CV, which is short and clearly shows the viewer’s perspective.

provides the best performance, and so we choose ResNet 152 for implementing an FCG ResNet encoder.

C. Decoder

We use beam search to select a sentence generated from the model with the highest probability within a preset beam width β , which is the product of probabilities of candidate words in a candidate sentence. Using beam search to produce a greedy approximation to the optimal solution is a standard approach to avoiding exponential blowups. Fig. 7 depicts the qualities of sentences with different beam widths. It is evident that choosing beam width $\beta = 16$ produces sentences that look more like comments, which also provides an acceptable trade-off of computation time.

D. Imperceptibility of the Generated Comments

Imperceptibility of generated comments is critical in providing covertness. We carry out intrinsic evaluation of imperceptibility using the following three metrics: BLEU (bilingual evaluation understudy) [37], ROUGE (recall-oriented understudy for gisting evaluation) [38], and PPL (perplexity) [39].

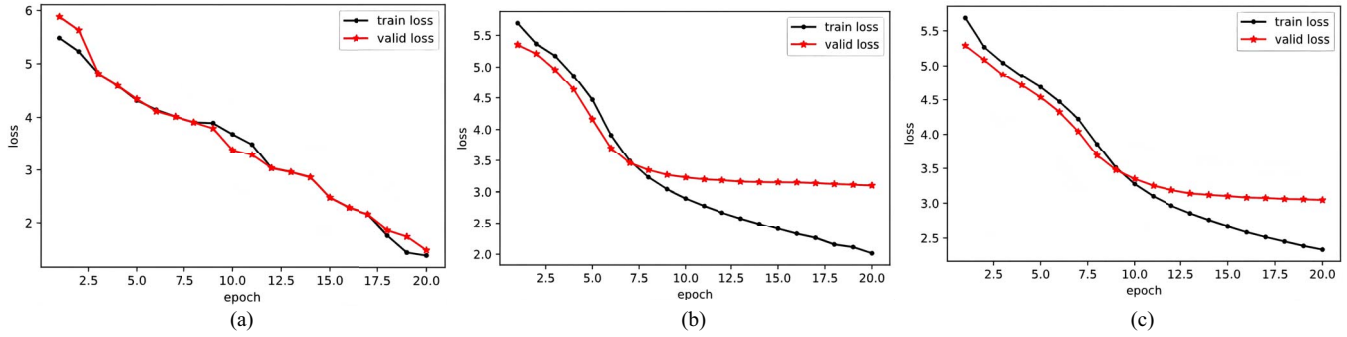


Fig. 6. When the number of epochs increases, the loss between the training set and validation set decreases, and ResNet 152 provides the best performance. (a) ResNet 152. (b) VGG 16. (c) Inception V3.






Image	$\beta = 1$	$\beta = 2$	$\beta = 4$	$\beta = 8$	$\beta = 16$
	The woman is holding a knife.	The woman with a knife to cook.	The woman with a knife to cook.	The knife looks sharp.	Ziqi sister who cooks with a knife is cool.
	There is grass and snow on the ground.	There is grass and snow on the ground.	Snow fell on the ground.	What cold the weather.	Snow fell on the ground.
	There are many yellow flowers on the ground.	Yellow bamboo.	The spring returns to the earth.	This scene is full of vitality.	The little yellow flowers are full of vitality.
	A group of people climb on a bridge.	The girl is dressed in red.	Ziqi sister went for a walk with a basket.	Ziqi sister went for a walk with a basket.	Little red riding hood.
	Red oil.	Chafing dish.	This chafing dish must be very spicy.	I love chafing dish.	This chafing dish must be very spicy.

Fig. 7. Qualities of sentences generated with different beam widths β .

1) *BLEU Comparison*: Let c be a generated comment and r_1, \dots, r_h the underlying reference comments in the dataset. Denote by $\text{BLEU-}l(c)$ ($l = 1, 2, 3, 4$) the ratio with the numerator being the number of l -grams that appear in c and in all r_i 's, and the denominator being the number of all l -grams in c , where an l -gram is l consecutive words in the underlying text.

We observe that generated comments that are short tend to have high BLEU scores. The reason is that l -grams in a short comment are more likely to appear in all the underlying references. To obtain a balanced measurement, we multiply the $\text{BLEU-}l(c)$ score with a punishment coefficient $\text{BP}(c)$ defined as follows:

$$\text{BP}(c) = \begin{cases} \exp(1 - l_r/l_c), & \text{if } l_c \leq l_r \\ 1, & \text{if } l_c > l_r \end{cases}$$

where l_c is the length of c and l_r the average length of the underlying references. Finally, we compute BLEU- l score as the average of $\text{BP}(c) \cdot \text{BLEU-}l(c)$ over all c 's.

FCG/ResNet-LSTM is built on neural image caption (NIC) [40] and sentence-by-sentence hiding (SSH) [41], where NIC is an award-winning model that generates image captions and SSH is an improved model of NIC. It is therefore reasonable to compare FCG with NIC and SSH. Table II depicts the BLEU- l scores over NIC, SSH, and FCG. It is evident that FCG is significantly better than both NIC and SSH on BLEU- l scores.

2) *ROUGE Comparisons*: Denote by $\text{ROUGE-}l(c)$ ($l = 1, 2$) the average of a ratio with respect to c for each reference r_i , where the numerator of the ratio is the number of l -grams that appear in both c and r_i , and the denominator is the number

TABLE II
BLEU SCORE COMPARISONS AT DIFFERENT BEAM WIDTHS

Model	Beam Width	BLEU-1	BLEU-2	BLEU-3	BLEU-4
NIC [40]	2	57.2	37.3	25.4	16.8
	4	58.3	39.3	26.0	17.3
	8	58.7	39.4	25.7	16.8
	16	59.1	39.8	26.0	16.9
SSH [41]	2	55.5	37.4	24.4	16.0
	4	56.1	37.6	24.8	16.6
	8	56.4	37.8	24.7	16.1
	16	56.6	37.9	24.7	16.1
FCG	2	70.0	50.9	37.1	27.2
	4	71.3	50.9	37.0	27.3
	8	71.5	51.1	36.8	27.3
	16	72.1	51.4	37.3	27.5

Note: Bold indicates the data with the best performance under the same experimental environment.

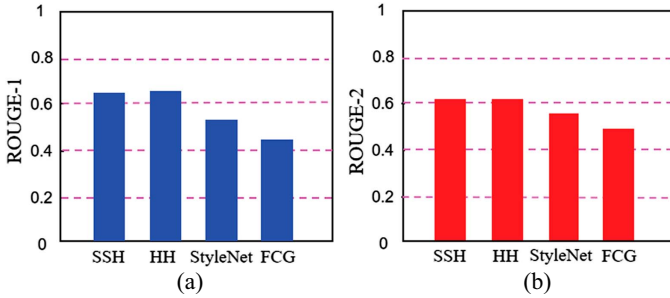


Fig. 8. ROUGE comparisons of different methods. (a) ROUGE-1. (b) ROUGE-2.

of all l -grams in r_i . ROUGE- l is the average of ROUGE- $l(c)$ over all c 's.

We compare FCG with the best-known image description models: SSH [41], hash hiding [41], and style network (StyleNet) [42]. In particular, StyleNet is a model to generate image captions with different styles that is more colloquial. Fig. 8 depicts the ROUGE-1 and ROUGE-2 scores over the union of Flickr8k and Bili2CV of these models. It can be seen that FCG is about 20% lower on ROUGE-1 and 16% lower on ROUGE-2, which are in favor of FCG, for they indicate that the comments generated by FCG are more diverse, and so are more in line with those made by diverse viewers.

3) *PPL Comparisons*: PPL measures the average number of words that can be encoded with a given BPW (bits per word), which is the average number of bits needed to encode one word. A comment-generating model that assigns a smaller PPL score to the comments in the test set means that it assigns a higher probability to the ground-truth comments, indicating that the model generates comments that would make more sense. We compare PPLs of FCG/ResNet-LSTM over the IMDB dataset [43] and a number of generative stego-methods, including LSTM [18], GS-VAE0/LSTM-LSTM (HC) [26], GS-VAE/BERT-LSTM (AC) [26], GS-VAE/BERT-LSTM (HC) [26], and GS-RNN (HC) [19], where AC and HC stand for conditional probability coding methods based on, respectively, arithmetic coding and Huffman coding. These models are based on the Croproc framework [19]. These models provide

TABLE III
PPL SCORES BY A NUMBER OF METHODS WITH VARIOUS BPWS OVER THE IMDB DATASET

LSTM [18]	BPW	1.000	2.000	3.000
	PPL	30.665	40.027	74.543
GS-VAE/LSTM-LSTM (HC) [26]	BPW	1.000	1.863	2.577
	PPL	45.115	49.511	59.532
GS-VAE/BERT-LSTM (AC) [26]	BPW	1.000	1.866	2.596
	PPL	30.266	36.349	40.832
GS-VAE/BERT-LSTM (HC) [26]	BPW	1.000	1.866	2.596
	PPL	30.266	36.349	40.832
GS-RNN (HC) [19]	BPW	1.000	1.845	2.565
	PPL	20.915	24.839	29.187
FCG/ResNet-LSTM	BPW	1.000	2.000	3.000
	PPL	20.771	22.133	24.851

Note: Bold indicates the data with the best performance under the same experimental environment.

high embedding rates and generate texts of satisfactory quality. Table III shows the PPL scores generated by these models under different BPWs. It follows from Table III that FCG/ResNet-LSTM substantially outperforms the other models.

E. Subjectivity

Among all models that generate image descriptions, we find that NIC [40] and StyleNet [42] generate image descriptions that are more in line with comments made by viewers. We compare the quality of comments generated by FCG, NIC, and StyleNet. A generated comment has a high quality if it cannot be distinguished from ground-truth comments. To ensure fairness, for each dataset of Flickr8k, Bili2CV, and Flickr8k \cup Bili2CV, we train an FCG model and select 500 generated comments and 500 ground-truth comments. We mix these 1000 comments for each dataset and ask human judges to score each comment in a 10-point scale, where 10 means that the judge considers the underlying comment a ground-truth comment with the highest confidence, while 1 means that the judge considers it a ground-truth comment with the least confidence. Fifty volunteers participated in the evaluation. For each comment over each dataset, we consider a comment a ground-truth comment if it receives an average score of seven or above, and consider it a generated comment otherwise.

Let TP denote the number of ground-truth comments correctly identified, TN the number of the non-ground-truth comments (i.e., the number of generated comments) correctly identified, FP the number of the generated comments incorrectly marked as ground-truth comments, and FN the number of ground-truth comments incorrectly marked as generated comments. Compute P_E as follows:

$$P_E = (P_{fa} + P_{md})/2$$

$$P_{fa} = FP/(TP + FP)$$

$$P_{md} = FN/(TP + FN)$$

where a larger P_E value indicates a higher quality of generated comments, meaning that it is more difficult to distinguish generated comments from ground-truth comments. Table IV shows the comparison results, from which it is evident that FCG is substantially better than both NIC and StyleNet.

TABLE IV
 P_E COMPARISONS OF DIFFERENT METHODS WITH DIFFERENT BEAM WIDTHS AND DIFFERENT BPWS

Method	Beam width		Flickr8k				Bili2CV				Flickr8k \cup Bili2CV			
NIC [40]	2	BPW	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000
		P_E	0.644	0.554	0.554	0.487	0.674	0.564	0.521	0.487	0.684	0.574	0.574	0.489
	8	BPW	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000
		P_E	0.674	0.524	0.604	0.447	0.671	0.601	0.592	0.498	0.691	0.688	0.574	0.504
	16	BPW	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000
		P_E	0.702	0.691	0.643	0.578	0.701	0.685	0.654	0.501	0.702	0.695	0.656	0.521
StyleNet [42]	2	BPW	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000
		P_E	0.632	0.621	0.602	0.541	0.604	0.588	0.543	0.522	0.642	0.640	0.601	0.598
	8	BPW	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000
		P_E	0.647	0.633	0.603	0.566	0.631	0.586	0.586	0.533	0.678	0.678	0.631	0.622
	16	BPW	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000
		P_E	0.708	0.690	0.649	0.588	0.697	0.658	0.602	0.589	0.732	0.666	0.666	0.641
FCG	2	BPW	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000
		P_E	0.788	0.771	0.767	0.811	0.767	0.811	0.776	0.801	0.756	0.783	0.783	0.821
	8	BPW	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000
		P_E	0.788	0.756	0.767	0.784	0.756	0.785	0.784	0.767	0.789	0.771	0.871	0.789
	16	BPW	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000	1.000	3.000	5.000	8.000
		P_E	0.802	0.805	0.799	0.821	0.802	0.796	0.787	0.821	0.806	0.806	0.811	0.825

Note: Bold indicates the data with the best performance under the same experimental environment.

F. Objectivity

We compute the differences of the spatial distributions between the generated comments and the ground-truth comments. To do so, we first convert each word in a comment into an embedding vector of a fixed length, then compute the central point of all word vectors contained in the comment to represent that comment. Finally, we project it to the two-dimensional space using the t-SNE algorithm [44]. Fig. 9 depicts the result on 600 ground-truth comments (the red dots) and 600 generated comments (the blue dots). It is evident that the reds and the blues are well blended, which is a clear indicator that the generated comments are imperceptible.

G. Feasibility and Efficiency

We present, through numerical experiments, concrete results of the following items:

- 1) the largest possible number of words we can expect to conceal in a 2-h video of 172 800 frames;
- 2) spread distributions, where a spread is the distance between two consecutive frame indexes in \mathbf{F} ;
- 3) the running time to generate a sequence of frame indexes \mathbf{F} from a sequence of word indexes, and we denote this time by T1;
- 4) the running time to compute \mathbf{J} from \mathbf{F} and we denote this time by T2.

Note that for CVF to be useful in practice, the value of d cannot be too small, and so we consider various settings of W - d matrices with $3 \leq d \leq 12$. Given a value of d , let $L(d)$ denote the maximum number of words that can be concealed in a two-hour video for randomly generated word indexes. Note that the upper bound of $L(d)$ is $u_d = (172\,800/15)/d = 11\,520/d$. For each value of d , we carry out five independent experiments, where in each experiment denoted by E_i ($1 \leq i \leq 5$), we generate at random a sequence of u_d word indexes using a

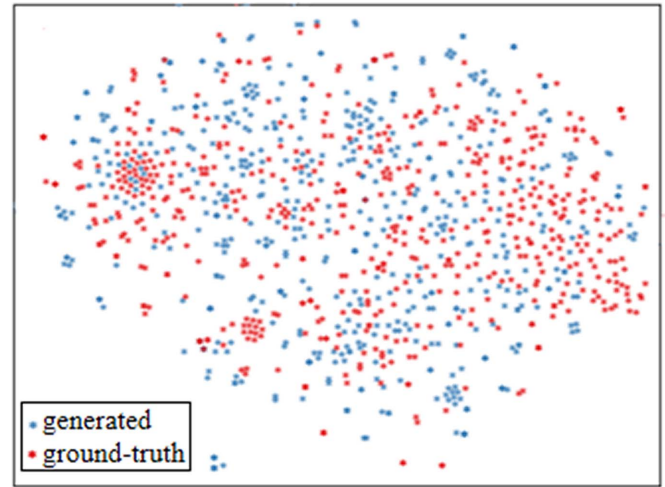


Fig. 9. Distributions of ground-truth and generated comments.

TABLE V
 EXPECTED MAXIMUM NUMBERS OF WORDS THAT CAN BE CONCEALED IN A TWO-HOUR VIDEO WITH VARIOUS VALUES OF d

d	3	4	5	6	7	8	9	10	11	12
E1	2958	2868	2488	2301	1972	1754	1490	1386	1289	1152
E2	2968	2899	2500	2318	1978	1754	1496	1389	1300	1169
E3	2978	2866	2498	2312	1974	1740	1498	1400	1289	1200
E4	2960	2852	2490	2321	1972	1749	1500	1389	1286	1166
E5	2974	2878	2494	2321	1971	1733	1496	1384	1284	1143
Avg	2968	2873	2494	2315	1973	1746	1496	1390	1290	1166

PRNG in the range of 1 to $2^{14} = 16\,384$, determine the maximum value of $L(d)$ by reducing u_d from trial and failure. The results are shown in Table V.

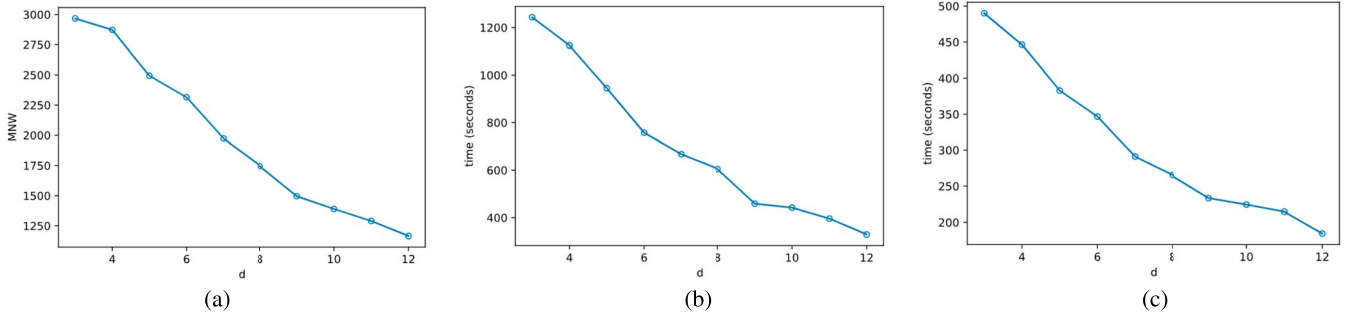


Fig. 10. Performance of CVF method for each value of d . (a) $L(d)$. (b) $T1(d)$. (c) $T2(d)$.

It is evident from Table V that the numbers in all experiments for each value of d are close, and Fig. 10(a) depicts the average number of words that can be concealed for each value of d , from which we can see that $L(d)$ is a nonlinear function of d .

For each value of d , we observe that there are spreads that reach the theoretical minimum of d and there are spreads that reach $2d$, where most spreads are closer to d than $2d$ —this seems reasonable for comments made by an active viewer.

Next, for each $L(d)$ determined above in each experiment E_i , we record $T1$ and $T2$. The values of $T1$, as expected, are close to each other in all experiments, so are those of $T2$. These experiments are carried out on a cloud server with a 2.3 GHz Xeon(R) Platinum 8260 CPU and 16 GB RAM. Fig. 10(b) and 10(c) depicts, respectively, the average $T1(d)$ and $T2(d)$ in each setting of d . We can see that both $T1(d)$ and $T2(d)$ seem to be functions of $L(d)$. Note that $T2(d)$ is substantially smaller than $T1(d)$. This is expected, for $T1$ involves generating a polynomial for each word index to form \mathbf{J} . It can be seen that $T1$ ranges from less than 6 min for $d = 12$ to about 20 min for $d = 3$ (note that a larger d means a smaller $L(d)$), and $T2$ ranges from about 3 min for $d = 12$ to about 8 min for $d = 3$. This is tractable for practical applications.

Finally, we generate \mathbf{W} - d matrices \mathbf{M} of dimension $\ell \times \ell$ with ℓ from 5 to 17 000, generate ℓ -dimensional vectors \mathbf{J} with elements between 1 and 16 384 pseudorandomly and compute $\mathbf{M} \cdot \mathbf{J} = \mathbf{F}$. We then verify if $\mathbf{M}^{-1} \cdot \mathbf{F} = \mathbf{J}$ with roundoff errors caused by floating-point entries in \mathbf{M}^{-1} . For each value of ℓ , we run the experiment 12 times, each with a different value of d between 1 and 12. Our experiments confirm that $\mathbf{M}^{-1} \cdot \mathbf{F} = \mathbf{J}$ in all cases.

VI. CONCLUSION AND FINAL REMARKS

CVF is the first-known generative method that leverages the live commenting feature on videos to achieve steganography. Shown to be covert, secure, efficient, and feasible, CVF can conceal secret messages of arbitrary lengths in one or more videos.

We believe that there is room for improving embedding capacity in videos. To this end, we would like to seek variants of \mathbf{W} - d matrices and explore new methods that can conceal more words than the current form of \mathbf{W} - d matrices. On the other hand, to preserve covertness for practical applications, we would also want to make sure that spreads are consistent with the norm.

To this end, it may be necessary to increase the value of d and use multiple videos to allow wider spreads. Another way to generate wider spreads to look more random is to generalize \mathbf{W} - d matrices by randomly generating the diagonal values between d and d' using a PRNG with sufficiently larger d and $d' > d$.

While it would deem sufficient for practical applications to find out the maximum number of words one can expect a two-hour video can conceal using numerical experiments, it would be interesting from a theoretical point of view to obtain an estimate mathematically in terms of d under the assumption that word indexes are uniformly distributed.

How to generate adequate distracting words for a given secret message is an interesting topic in its own right, which deserves a further investigation.

The ResNet-LSTM model we have trained for FCG treats frame images independently. However, comments made by viewers may relate to earlier frames. We believe that it is possible to combine advanced features of text-to-text transformers and image understanding models to generate comments that are aware of frame histories. Doing so makes live comments more attractive and less suspicious. This task calls for datasets much larger than Bili2CV.

REFERENCES

- [1] T. Y. Liu and W. H. Tsai, "A new steganographic method for data hiding in Microsoft word documents by a change tracking technique," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 1, pp. 24–30, Mar. 2007.
- [2] C. C. Chang, "Reversible linguistic steganography with Bayesian masked language modeling," *IEEE Trans. Comput. Social Syst.*, vol. 10, no. 2, pp. 714–723, Apr. 2023.
- [3] Y. Wang, W. Zhang, W. Li, X. Yu, and N. Yu, "Non-additive cost functions for color image steganography based on inter-channel correlations and differences," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2081–2095, 2020.
- [4] C. Yu, X. Zhang, X. Zhang, G. Li, and Z. Tang, "Reversible data hiding with hierarchical embedding for encrypted images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 2, pp. 451–466, Feb. 2022.
- [5] Y. Dong, X. H. Jiang, Z. H. Li, T. F. Sun, and Z. Z. Zhang, "Multi-channel HEVC steganography by minimizing IPM steganographic distortions," *IEEE Trans. Multimedia*, vol. 25, pp. 2698–2709, 2023.
- [6] Z. Li, X. Jiang, Y. Dong, L. Meng, and T. Sun, "An anti-steganalysis HEVC video steganography with high performance based on CNN and PU partition modes," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 1, pp. 606–619, Jan./Feb. 2023.
- [7] K. J. Chen et al., "Cover reproducible steganography via deep generative models," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 5, pp. 3787–3798, Sep./Oct. 2023.

- [8] B. Singh, A. Sur, and P. Mitra, "Steganalysis of digital images using deep fractal network," *IEEE Trans. Comput. Social Syst.*, vol. 8, no. 3, pp. 599–606, Jun. 2021.
- [9] C. F. Yang, X. Y. Luo, J. C. Lu, and F. L. Liu, "Extracting hidden messages of LSB steganography based on optimal stego subset," *Sci. China Inf. Sci.*, vol. 61, no. 11, pp. 237–239, Jan. 2018.
- [10] X. Chen and S. Chen, "Text coverless information hiding based on compound and selection of words," *Soft Comput.*, vol. 23, no. 15, pp. 6323–6330, Aug. 2019.
- [11] Y. Luo, J. Qin, X. Xiang, and Y. Tan, "Coverless image steganography based on multi-object recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 7, pp. 2779–2791, Jul. 2021.
- [12] K. Wu and C. Wang, "Steganography using reversible texture synthesis," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 130–139, Jan. 2015.
- [13] S. Li and X. Zhang, "Towards construction based data hiding: From secrets to fingerprint images," *IEEE Trans. Image Process.*, vol. 38, no. 3, pp. 1482–1497, Mar. 2019.
- [14] M. M. Liu, M. Q. Zhang, J. Liu, Y. N. Zhang, and Y. Ke, "Coverless information hiding based on generative adversarial networks," 2017, *arXiv:1712.06951*.
- [15] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li, "A novel image steganography method via deep convolutional generative adversarial networks," *IEEE Access*, vol. 6, pp. 38303–38314, 2018.
- [16] M. Chapman and G. Davida, "Hiding the hidden: A software system for concealing ciphertext as innocuous text," in *Proc. Int. Conf. Inf. Commun. Secur. (ICICS)*, in Lecture Notes in Computer Science, 1997, vol. 1334, no. 1, pp. 335–345.
- [17] P. Wayner, "Mimic functions," *Cryptologia*, vol. 16, no. 3, pp. 193–214, Jun. 1992.
- [18] T. Fang, M. Jaggi, and K. Argyraki, "Generating steganographic text with LSTMs," *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics Student Res. Workshop*, 2017, p. 100.
- [19] Z. L. Yang, X. Q. Guo, Z. M. Chen, Y. F. Huang, and Y. J. Zhang, "RNN-stega: Linguistic steganography based on recurrent neural networks," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1280–1295, May 2019.
- [20] L. Xiang, S. Yang, Y. Liu, Q. Li, and C. Zhu, "Novel linguistic steganography based on character-level text generation," *Mathematics*, vol. 8, no. 9, pp. 1558–1576, Jul. 2020.
- [21] X. Zhou et al., "Linguistic steganography based on adaptive probability distribution," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 5, pp. 2982–2997, Sep./Oct. 2022.
- [22] Z. Yang et al., "Graph-stega: Semantic controllable steganographic text generation guided by knowledge graph," 2020, *arXiv:2006.08339*.
- [23] Y. Li, J. Zhang, Z. Yang, and R. Zhang, "Topic-aware neural linguistic steganography based on knowledge graphs," *ACM/IMS Trans. Data Sci.*, vol. 2, no. 2, pp. 1–13, Apr. 2021.
- [24] M. Li, K. Mu, P. Zhong, J. Wen, and Y. Xue, "Generating steganographic image description by dynamic synonym substitution," *Signal Process.*, vol. 164, pp. 193–201, Nov. 2019.
- [25] P. Keserwani and P. P. Roy, "Text region conditional generative adversarial network for text concealment in the wild," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 3152–3163, May 2022.
- [26] Z. L. Yang, S. Y. Zhang, Y. T. Hu, Z. W. Hu, and Y. F. Huang, "VAE-stega: Linguistic steganography based on variational auto-encoder," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 880–895, 2021.
- [27] C. Wang, Y. Liu, Y. Tong, and J. Wang, "GAN-GLS: Generative lyric steganography based on generative adversarial networks," *CMC-Comput. Mater. Continua*, vol. 69, no. 1, pp. 1375–1390, Apr. 2021.
- [28] Y. Cao et al., "Generative steganography based on long readable text generation," *IEEE Trans. Comput. Social Syst.*, early access, May 19, 2022.
- [29] S. Ma, L. Cui, D. Dai, F. Wei, and X. Sun, "LiveBot: Generating live video comments based on visual and textual contexts," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 6810–6817.
- [30] B. Keith, *WordNet and Wordnets*. Amsterdam, The Netherlands: Elsevier, 2005. Accessed: May 13, 2022. [Online]. Available: <https://wordnet.princeton.edu/>
- [31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [32] M. Younis, W. Lalouani, N. Lasla, L. Emokpae and M. Abdallah, "Blockchain-enabled and data-driven smart healthcare solution for secure and privacy-preserving data access," *IEEE Syst. J.*, vol. 16, no. 3, pp. 3746–3757, Sep. 2022.
- [33] P. Bayar et al., "PPSF: A privacy-preserving and secure framework using blockchain-based machine-learning for IoT-driven smart cities," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2326–2341, Jul./Sep. 2021.
- [34] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *J. Artif. Intell. Res.*, vol. 47, pp. 853–899, May 2013.
- [35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.
- [37] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.
- [38] C. Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Text Summarization Branches Out*, 2004, pp. 74–81.
- [39] F. Jelinek, R. L. Mercer, L. R. Bahl, and K. Baker, "Perplexity—A measure of the difficulty of speech recognition tasks," *J. Acoust. Soc. Amer.*, vol. 62, no. S1, pp. S63–S63, 1977.
- [40] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3156–3164.
- [41] J. Wen, X. Zhou, M. Li, P. Zhong, and Y. Xue, "A novel natural language steganographic framework based on image description neural network," *J. Vis. Commun. Image Representation*, vol. 61, pp. 157–169, May 2019.
- [42] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng, "StyleNet: Generating attractive visual captions with styles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3137–3146.
- [43] A. Maas et al., "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics Human Lang. Technol.*, 2011, pp. 142–150.
- [44] H. Zhou, F. Wang, and P. Tao, "t-Distributed stochastic neighbor embedding method with the least information loss for macromolecular simulations," *J. Chem. Theory Comput.*, vol. 14, no. 11, pp. 5499–5510, Nov. 2018.



Yuling Liu received the B.E. and Ph.D. degrees in computer science from Hunan University, Changsha, China, in 2003 and 2008, respectively.

She is currently an Associate Professor and a Doctoral Supervisor with the College of Computer Science and Electronic Engineering, Hunan University. She was a Visiting Scholar with the University of Massachusetts, Lowell, MA, USA. Her current research interests include multimedia security, artificial intelligence security, steganography, and watermarking.



Cuilin Wang received the M.S. degree in software engineering from Hunan University, Changsha, China, in 2023.

She is currently working with Xiangxi Power Supply Branch, State Grid Hunan Electric Power Company, Ltd., Jishou, China. Her current research interests include multimedia security and steganography.



Jie Wang (Member, IEEE) received the Ph.D. degree in computer science from Boston University, Boston, MA, USA, in 1991.

He is a Professor of Computer Science with the University of Massachusetts, Lowell, MA, USA. He co-directs the Center for Internet Security and Forensics Education and Research, and chaired the Department of Computer Science from 2007 to 2016. His research interests include text AI applications, modeling and optimization algorithms, computational complexity, and computer security.

He has graduated over 20 Ph.D. students and involved in entrepreneurship.



Bo Ou received the B.S. and Ph.D. degrees in human-computer from Beijing Jiaotong University, Beijing, China, in 2008 and 2014, respectively.

He has been on the faculty of Hunan University (HNU), Changsha, China, where he is currently an Associate Professor. His current research interests include reversible data hiding and the corresponding researches.



Xin Liao (Senior Member, IEEE) received the B.E. and Ph.D. degrees in information security from Beijing University of Posts and Telecommunications, Beijing, China, in 2007 and 2012, respectively.

He is currently a Professor with Hunan University, Changsha, China. From 2016 to 2017, he was a Visiting Scholar with the University of Maryland, College Park, MD, USA. His current research interests include multimedia forensics, steganography, and watermarking.

Dr. Liao is an Associate Editor for the *IEEE Signal Processing Magazine*.