Reversible Linguistic Steganography With Bayesian Masked Language Modeling

Ching-Chun Chang

***Abstract*— Text authentication serves a vital role in the defense of digital identity and content against various types of cybercrime. The use of a digital signature is a common cryptographic technique for text authentication. Linguistic steganography can be applied to further conceal a digital signature within the corresponding text to facilitate data management. However, steganographic distortion lurking in the text, albeit almost imperceptible, has the potential to cause automatic computing machinery to make biased decisions. This has led to an interest in the pursuit of reversibility, the ability to reverse a steganographic process and remove distortion. In this article, we propose a reversible steganographic system for natural language text. We use a pre-trained transformer neural network for masked language modeling and embed messages in a reversible manner via predictive word substitution. Furthermore, we derive an adaptive steganographic route by taking account of predictive uncertainty, which is quantified based on a theoretical framework of Bayesian deep learning. Experimental results show that the proposed steganographic system can attain a proper balance between capacity, imperceptibility, and reversibility with close semantic and sentimental similarities between cover and stego texts.**

***Index Terms*— Bayesian uncertainty analysis, masked language modeling, reversible linguistic steganography.**

I. Introduction

AUTHENTICATION is an essential part of cyber security. It is the process of validating the identity of users and the integrity of digital content. As cyber space continues to expand in scope and scale, authentication plays an important role in maintaining trust against various types of deception, including, but not limited to, impersonating identities, spreading spam, disseminating fake news, sending malicious links, and tampering with digital media. A digital signature is a mathematical proof of authenticity using modern cryptographic techniques such as encryption and hashing [1]. The incorporation of a timestamp and other tamper-evident designs can further strengthen security. Such auxiliary data, however, carry the risk of accidental loss and mismanagement during storage, transmission, or format transformation.

Steganography is the practice of concealing information (e.g. a secret message, a copyright mark, or a serial number) within a carrier object [2]–[4]. Steganographic techniques have been applied to covert communications [5], ownership identification [6], broadcast monitoring [7], and traitor tracing [8]. It can also serve as an authentication solution by embedding auxiliary metadata in a digital file, thereby mitigating the risk of losing data. While steganographic distortion is often imperceptible to human sensory systems, it can bias decisions of automatic computing machinery. Previous studies of adversarial attacks have reported that machine learning models can be vulnerable to imperceptible perturbations [9]–[11]. Such distortion might also be inadmissible in some sensitive circumstances such as forensic science, legal proceedings, medical diagnosis, and military reconnaissance.

Reversibility is the key to removing steganographic distortion. Reversible computing describes the notion that a computational process can to some extent be time-reversible. Most reversible steganographic methods are designed for digital imagery [12]–[19], whereas methods for textual data remain relatively undeveloped. A possible explanation is that many well-developed tools are available for exploiting the redundancies in visual signals on which steganographic methods rely. In contrast, manipulating natural language texts can be much more challenging, considering that even the tiniest change in a character can be discernible to a careful reader. With the worldwide popularization of social media and the technological advances in natural language processing (NLP), textual data have become an important source of information. Hence, reversible steganography for textual data has emerged as a promising research field.

Typographical steganography is a methodology that embeds messages by manipulating the typeface, spacing, font size, or other typographical characteristics of texts [20]–[24]. It treats text documents as a special type of imagery and is usually applied to printed copies. However, portability and robustness are restricted because this class of method is unable to withstand retyping and font changes. Linguistic steganography deals instead with natural language *per se* and is thereby able to resist such text processing. It exploits linguistic knowledge and conceals messages by modifying linguistic properties, ideally without altering the sentence semantics or degrading sentence fluency. Linguistic steganography can be broadly categorized into the following classes: lexical class, syntactic class, and generative class. A typical lexical method is synonym substitution, which uses different synonyms of a word to represent different message digits [25]–[27]. In principle, words belonging to the same synonym set have similar meanings and thus can be substituted for

Manuscript received 9 December 2021; revised 26 January 2022 and 4 March 2022; accepted 18 March 2022. Date of publication 8 April 2022; date of current version 3 April 2023.

The author is with the Department of Computer Science, University of Warwick, Coventry, CV4 7AL, U.K. (e-mail: c.c.chang@warwickgrad.net).

Digital Object Identifier 10.1109/TCSS.2022.3162233

each other without causing notable semantic change. Syntactic methods are based on the fact that a sentence can be transformed into semantically equivalent syntactic structures such as active-to-passive voice conversion, subject–verb inversion, and topicalization [28]–[31]. Generative methods consider further that a sentence can be translated into a wide spectrum of forms or completely rewritten subject to minimal change to the semantics. A feasible approach is to use a set of machine translators to generate multiple translations of a given sentence [32]–[35]. Controllable natural language generation has also emerged as a promising direction for generative-level linguistic steganography [36]–[40]. Although these linguistic methods find their applications in areas such as secret communications and ownership identification, none of them meets the reversibility requirement for distortion-free text authentication.

In this article, we study reversible linguistic steganography. We apply a masked language model to generate a list of predictive words (Section II) and embed messages via predictive word substitution (Section III). The underlying principle is that most words can be retrieved within a finite number of predictive words, and this type of redundancy can be leveraged for reversibility. The performance of the steganographic system is associated with the accuracy of the predictive model. We further observe that the carrier words can be selected more efficiently by exploiting predictive uncertainty (Section IV). To determine an efficient route for message embedding, we study uncertainty quantification based on a theoretical framework of Bayesian deep learning (Section V).

The remainder of this article is organized as follows. Section II provides a brief overview of masked language modeling. Section III introduces a practical method of reversible linguistic steganography. Section IV offers a further discussion on steganographic routing. Section V presents Bayesian uncertainty quantification. Section VI shows the experimental results for the proposed systems. Concluding remarks are offered in Section VII.

II. Masked Language Modeling

Masked language modeling is a fill-in-the-blank task (a.k.a. a cloze test), where a model attempts to predict a masked word from the surrounding context words. Tokenization is a preliminary step whereby a text sequence is split into *tokens* (e.g. words, punctuation marks, and numbers). A masked language model takes an input sequence that contains one or more mask tokens and estimates the probability distribution of each mask token over the entire vocabulary.

Given a sequence of tokens, a fundamental issue across nearly all NLP tasks is how to represent tokens as numerical input in a computational model. The arguably simplest representation is the one-hot vector. It is a binary vector of $\|V\|$ elements, with all elements set to 0 except the element at the index of the corresponding word in a dictionary, where $\|V\|$ is the size of the vocabulary. This type of *denotational* representation is sparse and treats each word as a completely independent entity; hence, it cannot capture latent word semantics. In linguistics, the distributional hypothesis suggests that words tend to have similar meanings if occurring in similar contexts [41]. This gives rise to the notion of *distributional* representation. For a given corpus, a co-occurrence matrix of size $\|V\| \times \|V\|$ is computed by counting the number of times each word appears within a context window around the word of interest. Words with similar co-occurrence patterns are expected to have similar meanings. However, the matrix is of high sparsity due to the curse of dimensionality. While singular value decomposition (SVD) can be applied for dimensionality reduction, it has a disadvantage in terms of scalability. This algorithm is computationally expensive for large matrices and has to be performed again from scratch if the co-occurrence matrix is updated by incorporating new terms or corpora. A modern approach toward representation learning is to train a neural network model on a certain task (e.g. language modeling) with each word vector initialized randomly, and then update word vectors iteratively using a stochastic optimization algorithm (e.g. gradient descent). A common limitation of early word vectorization models (e.g. word2vec [42] and GloVe [43]) is their inapplicability to polysemy and homonymy. Multiple meanings of a word are conflated into a single indistinguishable representation regardless of the context within which the word appears.

Contextual word representation is an advanced concept in which each word vector can be adapted dynamically to its context [44]. The BERT model (standing for bidirectional encoder representations from transformers) is a state-of-the-art neural network developed by Google for learning contextual word representation [45]. The model is based on the transformer architecture which comprises multiple layers of self-attention modules and processes an input sequence bidirectionally [46]. To learn contextual word representation, the model is trained to perform masked language modeling. This task forces the model to learn more about contextual information. The knowledge learned from masked language modeling can then be transferred to tackle other downstream NLP tasks (e.g. machine translation, sentiment analysis, topic categorization). This is done by fine-tuning: choosing relevant layers in the pre-trained model, adding task-specific layers on top of the model, and training the model on new data. There are a variety of ways to use a pre-trained BERT model, and transfer learning is still a matter of ongoing research. Nevertheless, our steganographic system relies only on a basic function of the BERT model, namely, masked language modeling. The architectural details of the BERT masked language model are illustrated in Fig 1.

III. Reversible Linguistic Steganography

Reversible linguistic steganography considers the following scenario. A sender wishes to communicate a message to a receiver. The message is concealed in a carrier sequence of text, causing recoverable distortions to the carrier sequence. We refer to the original sequence as *cover* and its distorted counterpart as *stego*. The message varies with specific applications and is generally assumed to be a random binary sequence. The objective is to assure the accuracy of message extraction and text recovery, while keeping steganographic distortion as low as possible.

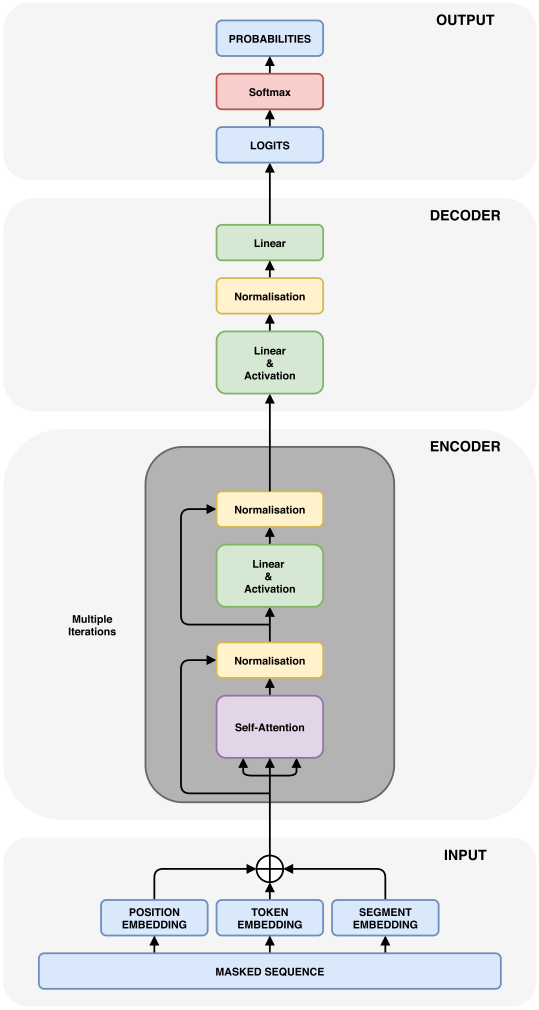


Fig. 1. Architecture of the BERT masked language model.

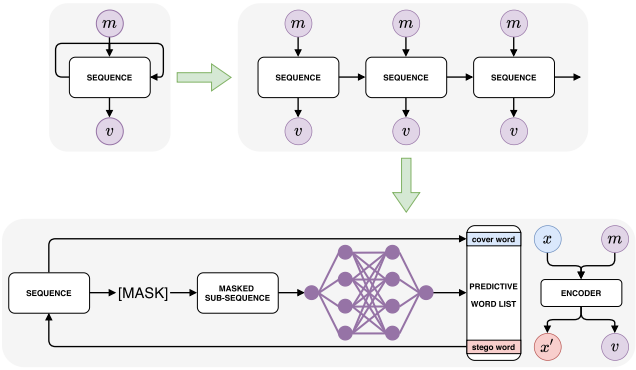


Fig. 2. Linguistic steganography based on the pre-trained masked language model.

Fig. 3. Carrier, noncarrier, and ambiguity intervals w.r.t. different settings of threshold.

The proposed method is based on masked language modeling, as depicted in Figure 2. To begin with, we mask a cover word in a text sequence and form a masked sub-sequence by taking the mask token and a fixed number of context words on the left-hand and right-hand sides. The masked sub-sequence is then fed into a pre-trained language model to obtain the probability distribution of the masked cover word. We sort the probabilities in descending order and derive a list of predictive words. The core idea is to substitute the cover word with a predictive word (or, more precisely, map the index of the cover word to another index) to represent a message digit.

Let us denote by x the index of the cover word in the list, where $x \in \mathbb{N}_0$ is a non-negative integer. We set a bound for the indices such that x must be less than the bound. If not, we skip the current word and proceed to the next cover word. We further set a threshold θ to separate the carrier and non-carrier indices. The carrier indices constitute a finite set of size $\|\theta\|$. To achieve reversibility, encoding must be a bijective function (i.e. a one-to-one correspondence). The number of unique combinations between all possible carrier indices and message digits is given by the Cartesian product of the two sets. Since all possible message digits form a binary set, message embedding doubles the size of the carrier set and causes an overlap between carrier and non-carrier indices. To avoid ambiguity in message extraction, the non-carrier indices have to be shifted outward. Index shifting can induce out-of-bound indices. The set of out-of-bound indices is also of size $\|\theta\|$. These indices are kept unshifted, and one flag bit is required to distinguish between a shifted and unshifted non-carrier index in the ambiguity interval $[(\text{bound} - \theta), (\text{bound} - 1)]$. The flag bits are regarded as an overhead of reversibility. At first glance, it seems that index shifting only acts to transfer the ambiguity interval from one end to another and serves little if any purpose. Yet, the overhead size is reduced substantially because the indices which are close to the bound rarely occur. Given a functional predictive model, we may reasonably assume that the frequency of an index value follows an exponential distribution (rather than a uniform distribution); that is, a smaller index occurs more frequently and vice versa. An illustration of carrier interval, non-carrier interval, and ambiguity interval with different θ settings is shown in Fig. 3. It can be seen that the maximum value of θ is equal to one-third of the bound. When θ is greater than this

Algorithm 1 Encoding

Input: x, $\boldsymbol{v}$, $\boldsymbol{m}$, count_m
Output: x', $\boldsymbol{v}$, count_m
▷ encoding
$\upsilon \leftarrow \varnothing$
if $x <$ bound **then**
  if $x < \theta$ **then** ▷ carrier zone
    $x' \leftarrow 2x + \boldsymbol{m}[\text{count}_m]$
    $\text{count}_m \leftarrow \text{count}_m + 1$
  else ▷ non-carrier zone
    $x' \leftarrow x + \theta$
  if $(\text{bound} - \theta) \leq x' <$ bound **then** ▷ ambiguity zone
    $\upsilon \leftarrow 0$
  else if $x' \geq$ bound **then** ▷ out of bound
    $x' \leftarrow x$
    $\upsilon \leftarrow 1$
else ▷ out of bound
  $x' \leftarrow x$
$\boldsymbol{v}.\text{append}(\upsilon)$ ▷ update flag list

Algorithm 2 Decoding

Input: x', $\boldsymbol{m}_{\text{rev}}$, $\boldsymbol{v}$, count_υ
Output: x, $\boldsymbol{m}_{\text{rev}}$, count_υ
▷ decoding
$m \leftarrow \varnothing$
if $x' <$ bound **then**
  if $x' < 2\theta$ **then** ▷ carrier zone
    $x \leftarrow \text{floor}(x'/2)$
    $m \leftarrow \text{mod}(x', 2)$
  else ▷ non-carrier zone
    $x \leftarrow x' - \theta$
  if $(\text{bound} - \theta) \leq x' <$ bound **then** ▷ ambiguity zone
    if $\boldsymbol{v}[\text{count}_\upsilon] = 1$ **then**
      $x \leftarrow x'$
    $\text{count}_\upsilon \leftarrow \text{count}_\upsilon - 1$
else ▷ out of bound
  $x \leftarrow x'$
$\boldsymbol{m}_{\text{rev}}.\text{append}(m)$ ▷ update reversed message list

value, the carrier interval overlaps with the ambiguity interval and there is no point in embedding one message bit at the cost of recording one flag bit.

A practical coding method is presented as follows. Based on the assumption about index frequency, we allocate a smaller amount of distortion to a smaller carrier index when embedding a digit. Let m be a binary message digit to be embedded. If x is within θ, we encode x and m into a stego index; otherwise, we shift x by θ

$$x' = \begin{cases} 2x + m & \text{if } x < \theta \\ x + \theta & \text{otherwise.} \end{cases} \tag{1}$$

If the stego index is out of bound, we reset it to its original value and record the cases by a flag bit

$$\upsilon = \begin{cases} 0 & \text{if } (\text{bound} - \theta) \leq x' < \text{bound} \\ 1 & \text{if } x' \geq \text{bound.} \end{cases} \tag{2}$$

Decoding is operated in a first-in last-out manner (i.e. in reverse order of encoding). In the decoding phase, a message bit is extracted by

$$m = \begin{cases} \text{mod}(x', 2) & \text{if } x' < 2\theta \\ \varnothing & \text{otherwise} \end{cases} \tag{3}$$

and an index is recovered by

$$x = \begin{cases} \text{floor}(x'/2) & \text{if } x' < 2\theta \\ x' - \theta & \text{otherwise.} \end{cases} \tag{4}$$

If x' is in the ambiguity interval, we read a flag bit to determine its original value. Pseudo-codes for the encoding and decoding procedures are provided in Algorithms 1 and 2.

IV. Steganographic Routing

Steganographic routing is the process of selecting a path for embedding a payload. Different paths can lead to different trade-off curves between capacity and distortion. Routing is particularly important in the case of a limited payload size because an optimal path can minimize distortion subject to a given payload constraint. There are basically two types of routing: static routing and dynamic routing. Static routing uses a default or manually configured path pre-shared between the encoder and the decoder. It is easy to implement, but cannot minimize distortion. Dynamic routing, on the other hand, constructs an adaptive path that reflects the degree of distortion caused by modifying each word. Recall that our coding design introduces a smaller degree of distortion to a smaller index and greater distortion to a larger index—that is to say, the degree of distortion is inversely proportional to predictive accuracy. Therefore, optimal routing is to select a path in descending order of predictive accuracy. While an optimal path is computable at the encoder, it cannot be reproduced at the decoder. The reason for this is that the word sequence used to derive the path is inconsistent with the word sequence received. A path is represented by a long sequence of digits, and storing such auxiliary information would be impractical. The problem of designing a dynamic path that is computable for both the encoder and the decoder is an intriguing one.

The most straightforward way to deal with textual data of a sequential nature is *sequential* routing. However, steganographic distortion imposed upon the preceding words would propagate, thereby impairing predictive performance on succeeding words. In other words, the contextual clues from the past are distorted and only the clues from the future remain intact. Introducing randomness is conducive to mitigating error propagation. Words are randomly selected for carrying the payload, thereby reducing the chance of encountering distorted context words. Furthermore, a random seed for initializing a pseudorandom number generator can serve as a secret key to enhance security. Nevertheless, the random variation on sequential routing is still a form of static routing and is not necessarily optimal. As previously mentioned, the optimal path is associated with predictive accuracy. The encoder and the

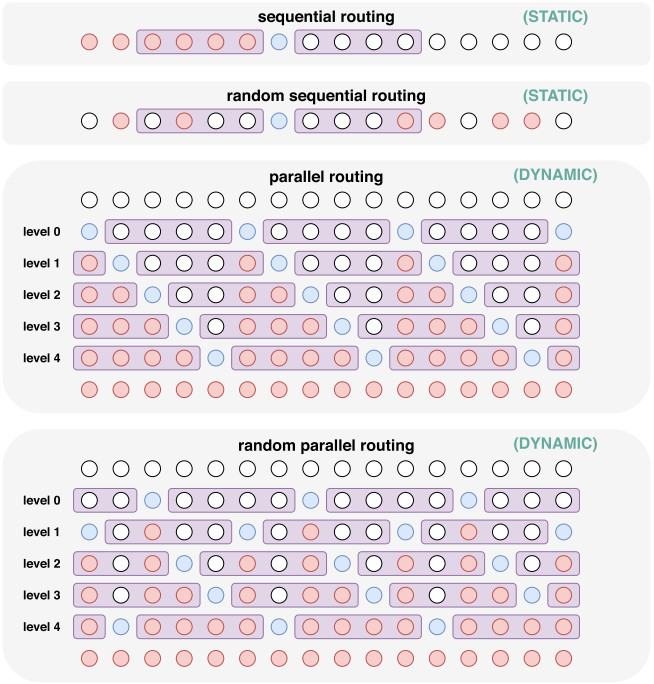


Fig. 4. Sequential and parallel routings with and without randomness.

decoder cannot derive the same optimal path because predictive accuracy is self-dependent—the accuracy of predicting a target word is related to the word itself. The target word must be kept unchanged to derive the same quantity; however, it has to be changed to carry information. These two objectives are mutually incompatible.

Predictive uncertainty is a concept closely related to predictive accuracy and depends purely on the context words. We can derive an alternative path in ascending order of uncertainty, assuming uncertainty is quantifiable. The synchronicity between the encoder and the decoder has to be ensured so that they can compute the same degree of uncertainty for each target word. For that reason, the context words have to be kept unchanged. This can be implemented by sampling target words at fixed intervals of context words. For instance, the context/target words are arranged in the following manner: a target word, a segment of context words, a target word, a segment of context words, and so forth. If the intended payload is beyond the capacity offered by the selected target words, we can select another set of target words by simply shifting the intervals, resulting in multi-level message embedding. The maximum number of levels is equal to the length of the context segment plus one. For each level, we can construct an adaptive path in ascending order of uncertainty. We refer to this method as *parallel* routing in the sense that a route is dynamically computed in each parallel level. Randomness can be introduced by randomly selecting a context/target pattern for each level. A variety of routing methods are illustrated in Fig. 4. Furthermore, we can set up an empirical threshold τ for filtering out words of high uncertainty to improve performance. In other words, when the prediction of a word is perceived to be highly uncertain, we keep the word completely intact to minimize distortion.

V. Bayesian Uncertainty Quantification

Most deep learning models are *deterministic* functions which offer only predictions without uncertainty information. Bayesian statistics offers a *probabilistic* interpretation of deep learning models from which the underlying uncertainty can be captured [47]. For a given masked sequence $\boldsymbol{s}$ and a training set $\mathcal{D}$, the predictive distribution of the masked word y is given by

$$p(y|\boldsymbol{s},\mathcal{D}) = \int \underbrace{p(y|\boldsymbol{s},\boldsymbol{\Phi})}_{\text{likelihood}} \underbrace{p(\boldsymbol{\Phi}|\mathcal{D})d\boldsymbol{\Phi}}_{\text{posterior}} \tag{5}$$

where $\boldsymbol{\Phi}$ denotes the model parameters. This can be interpreted as the average prediction over all plausible parameter settings according to the parameter posterior. For deep learning models, the derivation of the parameter posterior is analytically intractable; hence, we resort to variational inference to approximate the posterior by a variational distribution $q(\boldsymbol{\Phi})$, which belongs to a family of distributions of simpler form. By substituting the parameter posterior with the variational distribution and approximating the integral with Monte Carlo integration, we derive that

$$\begin{aligned} p(y|\boldsymbol{s},\mathcal{D}) &\overset{\text{VI}}{\approx} \int p(y|\boldsymbol{s},\boldsymbol{\Phi})q(\boldsymbol{\Phi})d\boldsymbol{\Phi} \\ &\overset{\text{MC}}{\approx} \frac{1}{T}\sum_{t=1}^{T} p\big(y|\boldsymbol{s},\hat{\boldsymbol{\Phi}}_t\big) \end{aligned} \tag{6}$$

where $\hat{\boldsymbol{\Phi}}_t \sim q(\boldsymbol{\Phi})$. Sampling model parameters from a variational distribution can be interpreted as *dropout* [48], which is a stochastic process of multiplying the output of each neurone by a random variable drawn from a Bernoulli distribution. Each dropout configuration corresponds to a plausible realization of a deep learning model with a portion of neurones deactivated. Applying T different dropout masks to the model is equivalent to performing stochastic forward passes for T repetitions, resulting in an ensemble of sparse neural network models. This process can be viewed as a proxy for a probabilistic deep learning model and is referred to as Monte Carlo dropout [49].

The predictive distribution is derived by averaging the likelihoods from T stochastic forward passes for each word

$$p(y=w_i|\boldsymbol{s},\mathcal{D}) \approx \frac{1}{T}\sum_{t=1}^{T} p\big(y=w_i|\boldsymbol{s},\hat{\boldsymbol{\Phi}}_t\big) \tag{7}$$

where w_i denotes the ith word in the dictionary. For masked language modeling, the likelihood is given by the normalized exponential function or softmax function

$$\begin{aligned} p\big(y=w_i|\boldsymbol{s},\hat{\boldsymbol{\Phi}}_t\big) &= \text{softmax}\big(f_i\big(\boldsymbol{s},\hat{\boldsymbol{\Phi}}_t\big)\big) \\ &= \frac{\exp\big(f_i\big(\boldsymbol{s},\hat{\boldsymbol{\Phi}}_t\big)\big)}{\sum_j \exp\big(f_j\big(\boldsymbol{s},\hat{\boldsymbol{\Phi}}_t\big)\big)} \end{aligned} \tag{8}$$

where f_i denotes the ith logit (i.e. raw prediction) from the model. In information theory, the uncertainty underlying a predictive distribution can be measured by Shannon entropy [50]

$$\mathbb{H} = -\sum_{i=1}^{\|V\|} p(y=w_i|\boldsymbol{s},\mathcal{D}) \ln p(y=w_i|\boldsymbol{s},\mathcal{D}). \tag{9}$$

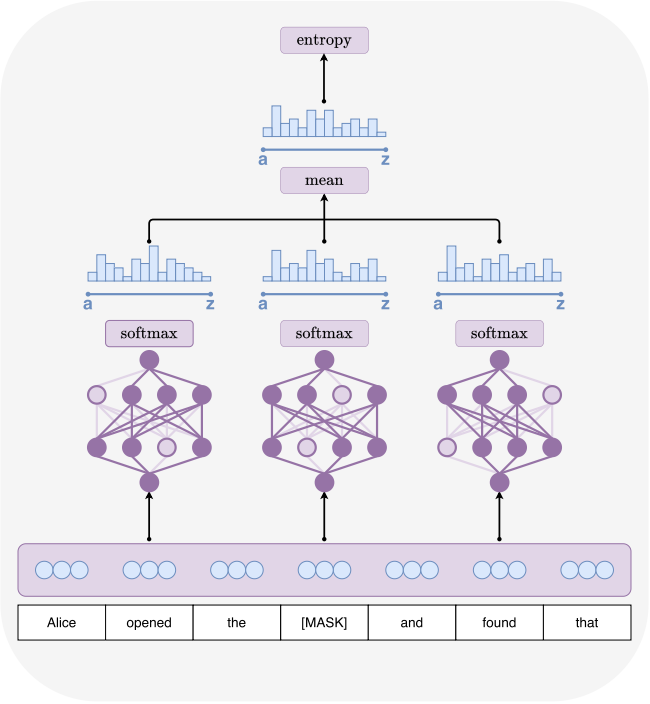


Fig. 5. Bayesian uncertainty quantification via Monte Carlo dropout.

Entropy is an encapsulation of information. It captures the average amount of information in predictive distribution. It is maximized when the predictive distribution is a uniform distribution; in other words, the model demonstrates the maximum uncertainty when each word is equally likely. It is minimized when only one word has a probability of 1 and all other words have a probability of 0. An overview of Bayesian uncertainty quantification is illustrated in Fig. 5.

VI. Experiments

We evaluate the proposed stego system with different settings of bound and threshold and compare different routings of interest. The trade-off between capacity, imperceptibility, and reversibility is examined with additional analysis on the semantic and sentimental similarities between cover and stego texts. Furthermore, a discussion on possible improvements is provided.

A. Experimental Setup

Our cover text consists of 8 selected paragraphs from a work of classic English literature, *Alice's Adventure in Wonderland*. The text contains 711 words plus punctuation. Each letter is made lower case. The BERT model has a vocabulary size of 30 522 tokens. The number of context words on each side of the target word (i.e. the length of the context segment) is set to 32 so that each input masked sub-sequence has 65 words. The number of Monte Carlo dropout samples (i.e. stochastic forward passes) is set to 1000. The predictive accuracy of the pre-trained BERT model can be represented by probability distribution function (PDF) and cumulative distribution function (CDF) of the word index, as shown in Fig. 6. It suggests

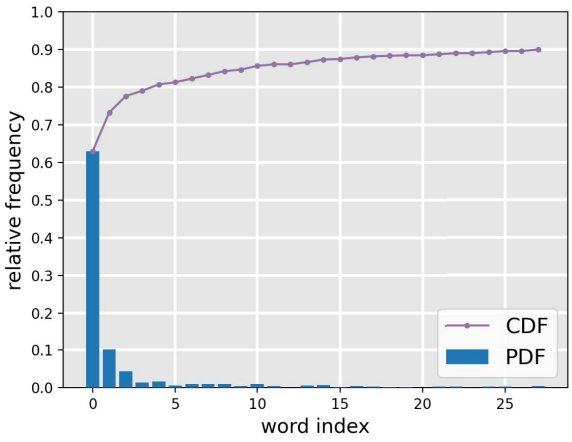


Fig. 6. Distribution of word index.

that more than 60% of words can be accurately predicted and about 90% of words are among the top 25 predictions in the case of no steganographic distortion.

B. System Evaluation

We evaluate our steganographic system with different settings of bound and θ and analyze the trade-off between capacity, imperceptibility, and reversibility. Capacity is measured by the number of payload bits (absolute value) and the payload bits per word (relative value). Imperceptibility is measured by the cosine similarity between the cover sequence and the stego sequence in the vector representation space. Reversibility is not an all-or-nothing proposition. We quantify reversibility by the number of flag bits used to disambiguate colliding word indices (the lower the better). Figs. 7 and 8 show, respectively, the capacity–imperceptibility curves and the capacity–reversibility curves from different routing methods. There is a general trend of decreasing imperceptibility and reversibility with increasing capacity. For the same threshold ($\theta = 1$), a smaller bound preserves a better similarity because there are fewer non-carrier words to be shifted. On the other hand, a smaller bound incurs more flag bits because ambiguous indices, which are close to the bound, appear more frequently. When the threshold is raised to the maximum ($\theta = \text{bound}/3$), the reachable capacity is increased. In addition, increased imperceptibility is obtained at the expense of lower reversibility. A comparison between parallel routing and sequential routing confirms the advantage of dynamic strategy over static strategy. The parallel routing takes account of predictive uncertainty and constructs an adaptive path for message embedding. The results also suggest that the random method has a positive effect on both the routing methods. Furthermore, when a few carrier words are filtered out by their uncertainty magnitudes (with a threshold τ), the reachable capacity is reduced and yet greater imperceptibility and reversibility are achieved. A more advanced uncertainty analyzer is expected to further improve performance. We would also like to point out that the current uncertainty analyzer requires computationally expensive Monte Carlo dropout. To facilitate real-time applications, a more efficient way to estimate uncertainty has yet to be developed.

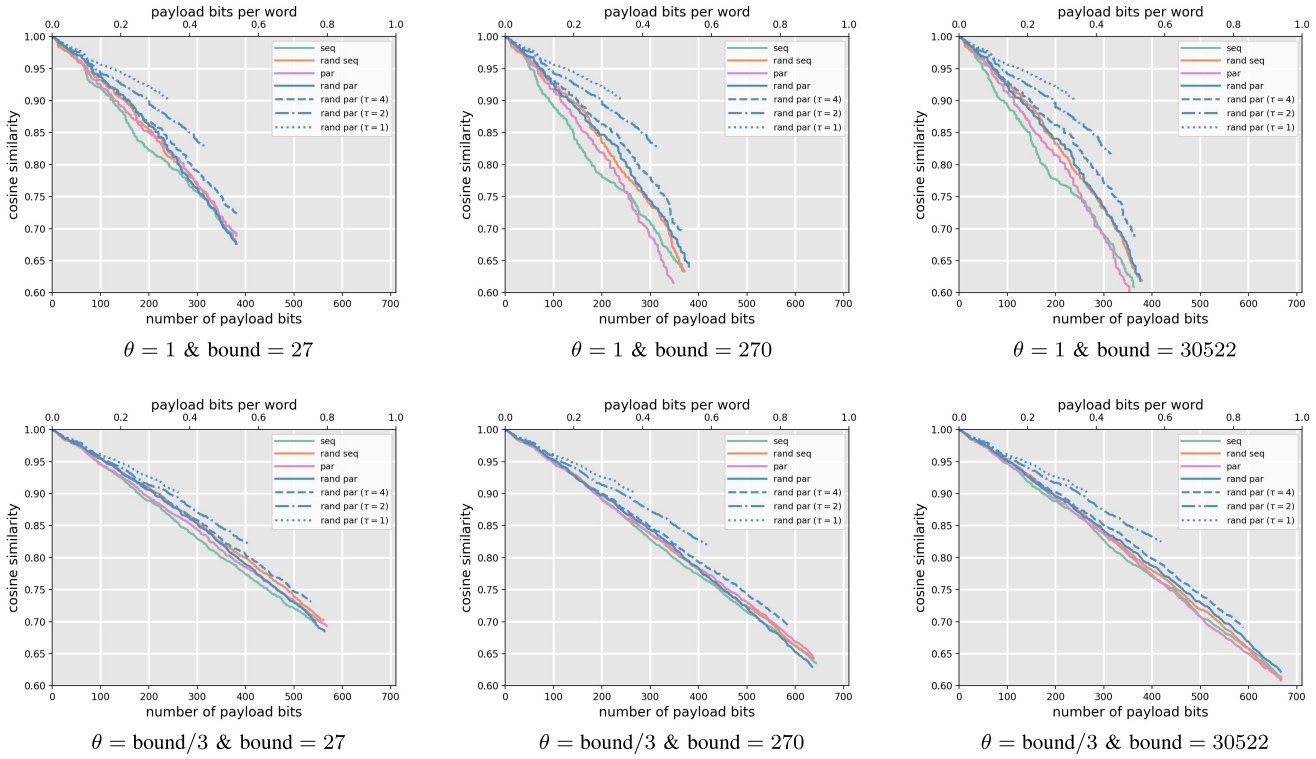


Fig. 7. Capacity–imperceptibility curves of different routings w.r.t. various settings of bound and threshold.

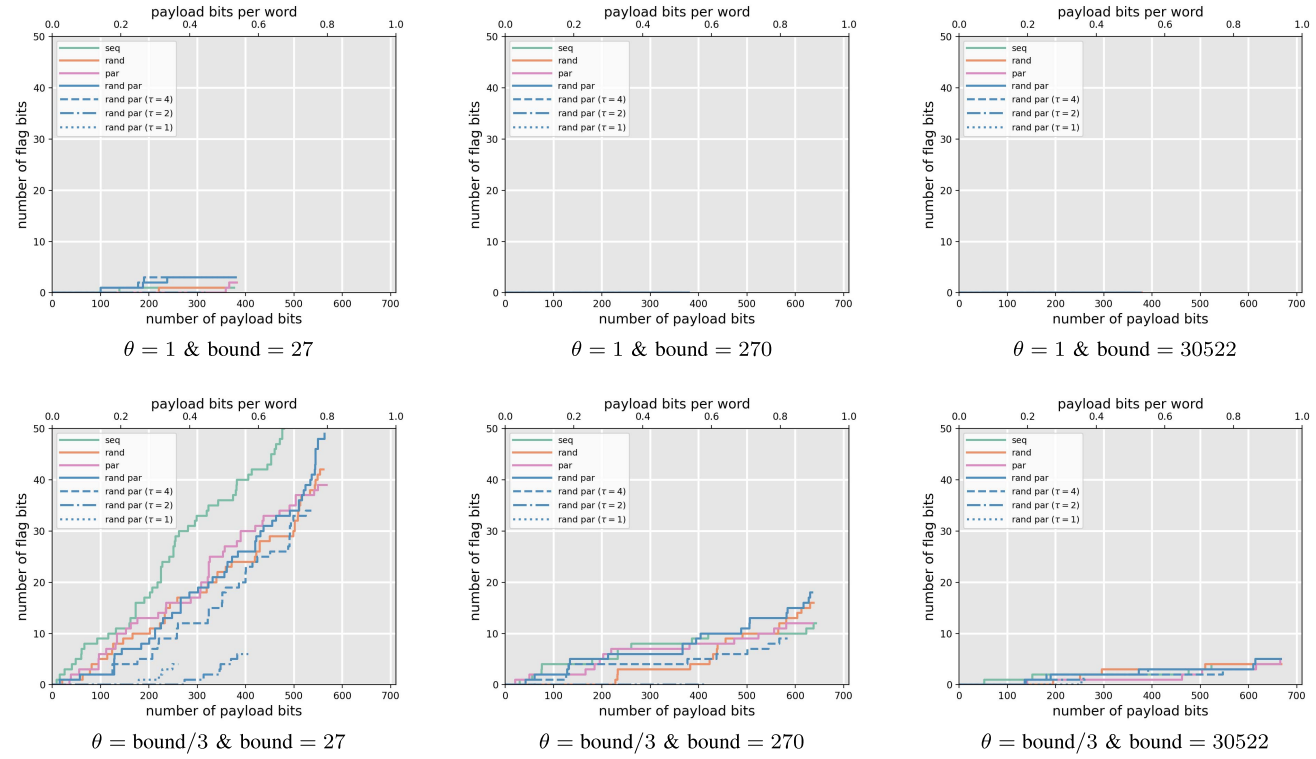
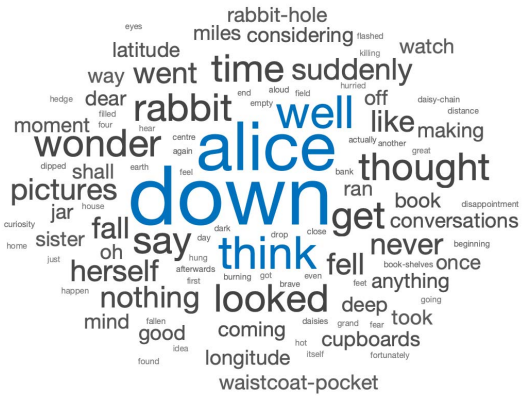


Fig. 8. Capacity–reversibility curves of different routings w.r.t. various settings of bound and threshold.

C. Semantics Analysis

Figs. 9 and 10 display a part of the cover text and the corresponding stego text, along with the word clouds for visualizing the frequency distributions of cover words and stego words. The stego text is generated by the random parallel routing with $\theta = 1$, bound $= 270$ and $\tau = 1$. We set the capacity to 0.3 bits per word so that more than 200 payload bits are embedded. This number is arguably sufficient for



alice was beginning to get very tired of sitting by her sister on the bank , and of having nothing to do: once or twice she had peeped into the book her sister was reading , but it had no pictures or conversations in it , “and what is the use of a book , ” thought alice “without pictures or conversations?”

so she was considering in her own mind (as well as she could , for the hot day made her feel very sleepy and stupid) , whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies , when suddenly a white rabbit with pink eyes ran close by her .

there was nothing so very remarkable in that; nor did alice think it so very much out of the way to hear the rabbit say to itself , “oh dear ! oh dear ! i shall be late ! ” (when she thought it over afterwards , it occurred to her that she ought to have wondered at this , but at the time it all seemed quite natural) ; but when the rabbit actually took a watch out of its waistcoat-pocket , and looked at it , and then hurried on , alice started to her feet , for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket , or a watch to take out of it , and burning with curiosity , she ran across the field after it , and fortunately was just in time to see it pop down a large rabbit-hole under the hedge .

in another moment down went alice after it , never once considering how in the world she was to get out again .

the rabbit-hole went straight on like a tunnel for some way , and then dipped suddenly down , so suddenly that alice had not a moment to think about stopping herself before she found herself falling down a very deep well .

Fig. 9. Word cloud and sample paragraphs of cover text with highlighted carrier words (in blue) and non-carrier words (in green).

alice were beginning to grow very tired , sitting near their sister on the bank , and of having nothing to do: once or once she had peeped into the book her sister was reading - but it had no photos and conversations on it) “and what would the use for a book , ” thought alice “without pictures and conversations?”

so she kept considering in her own mind (as well if it could , for the hot day made her look very sleepy and stupid) , whether the pleasure in making a daisy-chain would seem worth her trouble of getting up and picking the daisies , when suddenly a white rabbit without pink eyes ran close by her .

it was nothing so very remarkable in that; nor would alice think it so very much out of the way and hear the rabbit say to itself , “oh god ! oh no ! i shall be late ! ” (when she thought it over afterwards , something occurred to me that she had to have wondered at this ; but for the time it all seemed quite natural enough ; but when a rabbit actually took a watch out of its waistcoat-pocket - and looked at it ... and then hurried on . alice started to her feet , then it flashed across her mind that alice has never before seen any rabbit with either a waistcoat-pocket , or a watch - take out of it , and burning in curiosity , alice ran across the field after one , and fortunately was just in line to see one pop down a large rabbit-hole under some hedge .

in another moment down went alice after it , never once considering how in this world she was to get out again ;

that rabbit-hole went straight on like the tunnel for some way , and then dipped suddenly down , such suddenly that alice had not a moment of think about stopping herself because she felt herself falling in a very deep well .

Fig. 10. Word cloud and sample paragraph of stego text with highlighted carrier words (in blue) and non-carrier words (in green).

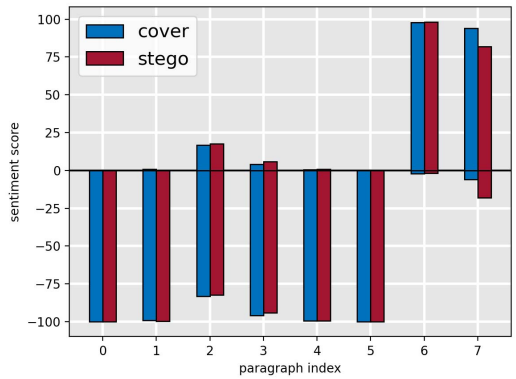


Fig. 11. Sentiment analysis of cover and stego texts.

many authentication applications. Perfect reversibility is also guaranteed without any overhead information. We can observe that the cover text and stego text are similar in terms of the semantics and the frequency distribution of words. However, closer inspection of the stego text shows that there are some unnatural word usages and grammatical mistakes. A possible refinement may be made by filtering out grammar words and named entities and manipulating content words only. Furthermore, a carefully designed word checker could be used to regulate the manipulations.

D. Sentiment Analysis

We carry out a sentiment analysis on a stego text generated using the aforementioned configurations. Fig. 11 reveals the positive/negative sentiment scores for each cover paragraph and each stego paragraph. The scores are obtained from a transformer-based sentiment analyzer. It is observed that the cover text and the stego text have very similar sentiment patterns, suggesting that steganographic distortion only produces minimal fluctuations in sentence sentiment. For particular sentiment-oriented applications, one may refine the system by retaining some salient contributory words which have a dominant influence upon text sentiment.

VII. Conclusion

In this work, we introduce a linguistic stego system with reversibility based on predictive word substitution. We use a pre-trained masked language model to generate a list of predictive words and embed a message digit by replacing the target word with one of the predictive words. The underlying assumption of the reversible coding is that word indices follow approximately an exponential distribution. We further apply a theoretical framework of Bayesian deep learning to quantify the uncertainty in the masked language model and use it to determine an adaptive route for message embedding. Our stego system achieves perfect reversibility without extra auxiliary information under limited capacity conditions. It also maintains close vector space, and semantic and sentimental similarities between cover and stego texts. Imperceptibility analysis suggests that steganographic distortion is to some extent indiscernible in a computing sense. In reality, however, even an extra punctuation mark or unusual collocation may be noted by a careful reader. Therefore, further improvement in imperceptibility is required. We also envisage further progress in uncertainty analysis such that the computational efficiency meets real-time requirements. Furthermore, while the proposed stego system is based primarily on lexical substitution, syntactic and generative methods also deserve investigation. We hope this article can shed light on future research devoted to reversible linguistic steganography.

References

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.
[2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Syst. J.*, vol. 35, nos. 3–4, pp. 313–336, 1996.
[3] R. J. Anderson and F. A. P. Petitcolas, "On the limits of steganography," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 474–481, 1998.
[4] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding—A survey," *Proc. IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.
[5] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, "Writing on wet paper," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3923–3935, 2005.
[6] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Process.*, vol. 6, no. 12, pp. 1673–1687, 1997.
[7] G. Depovere *et al.*, "The VIVA project: Digital watermarking for broadcast monitoring," in *Proc. Int. Conf. Image Process. (ICIP)*, Kobe, Japan, 1999, pp. 202–205.
[8] S. He and M. Wu, "Joint coding and embedding techniques for multimedia fingerprinting," *IEEE Trans. Inf. Forensics Secur.*, vol. 1, no. 2, pp. 231–247, 2006.
[9] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Banff, AB, Canada, 2014, pp. 1–10.
[10] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015, pp. 1–11.
[11] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 86–94.
[12] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication," *Proc. SPIE*, vol. 4314, pp. 197–208, 2001.
[13] C. De Vleeschouwer, J.-F. Delaigle, and B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management," *IEEE Trans. Multimedia*, vol. 5, no. 1, pp. 97–105, 2003.
[14] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, 2005.
[15] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, 2007.
[16] S. Lee, C. D. Yoo, and T. Kalker, "Reversible image watermarking based on integer-to-integer wavelet transform," *IEEE Trans. Inf. Forensics Secur.*, vol. 2, no. 3, pp. 321–330, 2007.
[17] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 989–999, 2009.
[18] C. Qin, C.-C. Chang, Y.-H. Huang, and L.-T. Liao, "An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 7, pp. 1109–1118, 2013.
[19] C.-C. Chang, "Adversarial learning for invertible steganography," *IEEE Access*, vol. 8, pp. 198425–198435, 2020.
[20] S. H. Low, N. F. Maxemchuk, J. T. Brassil, and L. O'Gorman, "Document marking and identification using both line and word shifting," in *Proc. IEEE Conf. Compt. Commun. (INFOCOM)*, vol. 2, Boston, MA, USA, 1995, pp. 853–860.
[21] S. H. Low, N. F. Maxemchuk, and A. M. Lapone, "Document identification for copyright protection using centroid detection," *IEEE Trans. Commun.*, vol. 46, no. 3, pp. 372–383, 1998.

[22] J. T. Brassil, S. H. Low, and N. F. Maxemchuk, "Copyright protection for the electronic distribution of text documents," *Proc. IEEE*, vol. 87, no. 7, pp. 1181–1196, 1999.
[23] L. Y. Por, T. F. Ang, and B. Delina, "WhiteSteg: A new scheme in information hiding using text steganography," *WSEAS Trans. Comp.*, vol. 7, no. 6, pp. 735–745, 2008.
[24] C. Xiao, C. Zhang, and C. Zheng, "FontCode: Embedding information in text documents using glyph perturbation," *ACM Trans. Graph.*, vol. 37, no. 2, pp. 1–16, 2018.
[25] K. Winstein, "Lexical steganography through adaptive modulation of the word choice hash," unpublished.
[26] I. A. Bolshakov, "A method of linguistic steganography based on collocationally-verified synonymy," in *Proc. Int. Workshop Inf. Hiding (IH)*, Toronto, ON, Canada, 2005, pp. 180–191.
[27] C.-Y. Chang and S. Clark, "Practical linguistic steganography using contextual synonym substitution and a novel vertex coding method," *Comput. Linguistics*, vol. 40, no. 2, pp. 403–448, 2014.
[28] M. Topkara, U. Topkara, and M. J. Atallah, "Words are not enough: Sentence level natural language watermarking," in *Proc. ACM Int. Workshop Contents Protection Secur. (MCPS)*, Santa Barbara, CA, USA, 2006, pp. 37–46.
[29] B. Murphy and C. Vogel, "The syntax of concealment: Reliable methods for plain text information hiding," *Proc. SPIE*, vol. 6505, pp. 351–362, 2007.
[30] H. M. Meral, B. Sankur, A. S. Özsoy, T. Güngör, and E. Sevinç, "Natural language watermarking via morphosyntactic alterations," *Comput. Speech Lang.*, vol. 23, no. 1, pp. 107–125, 2009.
[31] C.-Y. Chang and S. Clark, "The secret's in the word order: Text-to-text generation for linguistic steganography," in *Proc. Int. Conf. Comput. Linguistic (COLING)*, Mumbai, India, 2012, pp. 511–528.
[32] C. Grothoff, K. Grothoff, L. Alkhutova, R. Stutsman, and M. Atallah, "Translation-based steganography," in *Proc. Int. Workshop Inf. Hiding (IH)*, Barcelona, Spain, 2005, pp. 219–233.
[33] R. Stutsman, C. Grothoff, M. Atallah, and K. Grothoff, "Lost in just the translation," in *Proc. ACM Symp. Appl. Comput. (SAC)*, Dijon, France, 2006, pp. 338–345.
[34] P. Meng, L. Hang, Z. Chen, Y. Hu, and W. Yang, "STBS: A statistical algorithm for steganalysis of translation-based steganography," in *Proc. Int. Workshop Inf. Hiding (IH)*, Calgary, AB, Canada, 2010, pp. 208–220.
[35] C.-Y. Chang and S. Clark, "Adjective deletion for linguistic steganography and secret sharing," in *Proc. Int. Conf. Comput. Linguistic (COLING)*, Mumbai, India, 2012, pp. 493–510.
[36] P. Wayner, "Strong theoretical stegnography," *Cryptologia*, vol. 19, no. 3, pp. 285–299, 1995.
[37] M. Chapman and G. I. Davida, "Hiding the hidden: A software system for concealing ciphertext as innocuous text," in *Proc. Int. Conf. Inf. Commun. Secur. (ICICS)*, Beijing, China, 1997, pp. 335–345.
[38] Z. Yang, X. Guo, Z. Chen, Y. Huang, and Y. Zhang, "RNN-Stega: Linguistic steganography based on recurrent neural networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 5, pp. 1280–1295, 2019.
[39] Z. Yang, S. Zhang, Y. Hu, Z. Hu, and Y. Huang, "VAE-Stega: Linguistic steganography based on variational auto-encoder," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 880–895, 2021.
[40] S. Zhang, Z. Yang, J. Yang, and Y. Huang, "Linguistic steganography: From symbolic space to semantic space," *IEEE Signal Process. Lett.*, vol. 28, pp. 11–15, 2021.
[41] Z. S. Harris, "Distributional structure," *Word*, vol. 10, nos. 2–3, pp. 146–162, 1954.
[42] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Scottsdale, AZ, USA, 2013, pp. 1–12.
[43] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Linguistics (EMNLP)*, 2014, pp. 1532–1543.
[44] M. E. Peters *et al.*, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics (NAACL)*, New Orleans, LA, USA, 2018, pp. 2227–2237.
[45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics (NAACL)*, Minneapolis, MN, USA, 2019, pp. 4171–4186.
[46] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, Long Beach, CA, USA, 2017, pp. 1–11.
[47] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 2016.
[48] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.
[49] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, 2016, pp. 1050–1059.
[50] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

Ching-Chun Chang received the Ph.D. degree in computer science from the University of Warwick, Coventry, U.K., in 2019. He participated in a short-term scientific mission supported by European Cooperation in Science and Technology Actions with the Faculty of Computer Science, Otto von Guericke University Magdeburg, Germany, in 2016. He was granted the Marie-Curie Fellowship and participated in a research and innovation staff exchange scheme supported by Marie Skłodowska-Curie Actions with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, USA, in 2017. He was a Visiting Scholar with the School of Computer and Mathematics, Charles Sturt University, Australia, in 2018, and the School of Information Technology, Deakin University, Australia, in 2019. He was a Research Fellow with the Department of Electronic Engineering, Tsinghua University, China, in 2020. He has been a Post-Doctoral Fellow with the National Institute of Informatics, Japan, since 2021. His research interests include steganography, watermarking, forensics, biometrics, cybersecurity, applied cryptography, image processing, computer vision, natural language processing, computational linguistics, machine learning, and artificial intelligence.