A Content-Preserving Secure Linguistic Steganography

Lingyun Xiang[1], Chengfu Ou[2], Xu He[1], Zhongliang Yang[3*], Yuling Liu[4]

[1]School of Computer Science and Technology, Changsha University of Science and Technology
[2]College of Cyberspace Security, Jinan University
[3]School of Cyberspace Security, Beijing University of Posts and Telecommunications
[4]College of Cyber Science and Technology, Hunan University
xiangly@csust.edu.cn, hahally@stu2025.jnu.edu.cn, hexu2345@gmail.com, yangzl@bupt.edu.cn, yuling_liu@hnu.edu.cn

Abstract

Existing linguistic steganography methods primarily rely on content transformations to conceal secret messages. However, they often cause subtle yet looking-innocent deviations between normal and stego texts, posing potential security risks in real-world applications. To address this challenge, we propose a content-preserving linguistic steganography paradigm for perfectly secure covert communication without modifying the cover text. Based on this paradigm, we introduce CLstega (*C*ontent-preserving *L*inguistic *stega*nography), a novel method that embeds secret messages through controllable distribution transformation. CLstega first applies an augmented masking strategy to locate and mask embedding positions, where MLM(masked language model)-predicted probability distributions are easily adjustable for transformation. Subsequently, a dynamic distribution steganographic coding strategy is designed to encode secret messages by deriving target distributions from the original probability distributions. To achieve this transformation, CLstega elaborately selects target words for embedding positions as labels to construct a masked sentence dataset, which is used to fine-tune the original MLM, producing a target MLM capable of directly extracting secret messages from the cover text. This approach ensures perfect security of secret messages while fully preserving the integrity of the original cover text. Experimental results show that CLstega can achieve a 100% extraction success rate, and outperforms existing methods in security, effectively balancing embedding capacity and security.

Extended version — https://arxiv.org/abs/2511.12565

Introduction

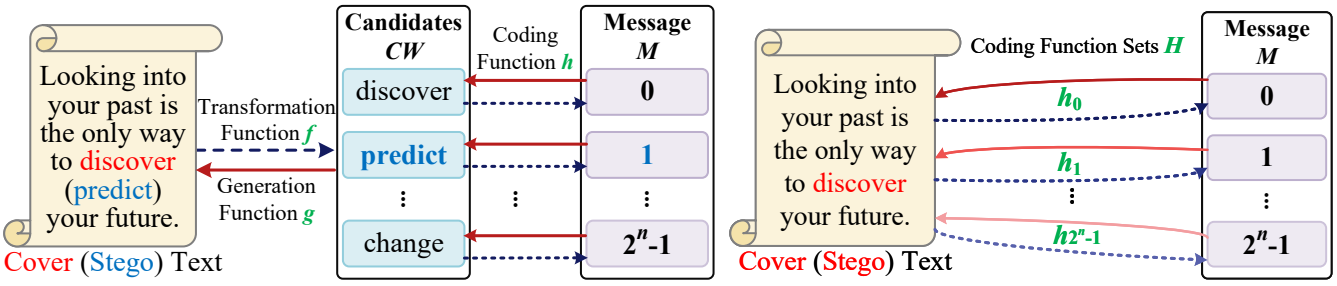
Steganography (Kahn 1996) is a technique that conceals secret messages in natural covers such as image (Hu et al. 2023), video (Mao et al. 2024), audio (Su et al. 2024), and text (Ding et al. 2023a) in an imperceptible manner. Its core objective is to hide the existence of secret messages under third-party surveillance, thereby ensuring the secure transmission of that message (Simmons 1984). Among various cover types, natural language is one of the most commonly used for message concealment in everyday communication (Zhang, Liu, and Zhang 2024), due to the advantage of high efficiency in the data transmission process (Yi et al. 2022). This makes linguistic steganography (LS), which employs natural language as a cover, increasingly popular in recent years (Idres and Yaseen 2023).

Prior studies primarily focus on content-transformation-based linguistic steganography, which can be divided into two categories: modification-based linguistic steganography (MLS) (Ueoka, Murawaki, and Kurohashi 2021; Zheng and Wu 2022; Xiang et al. 2023) and generation-based linguistic steganography (GLS) (Ziegler, Deng, and Rush 2019; Shen, Ji, and Han 2020; Zhang et al. 2021; Ding et al. 2023b). However, these methods struggle to significantly eliminate the deviation in statistical, semantic, and perceptual due to the existence of awkward content transformation manipulations (inappropriate word selection and unnatural syntactic transformation) during embedding secret messages, resulting in some potential clues to steganalysis methods(Yang et al. 2020a, 2021; Peng et al. 2023; Xue et al. 2023; You et al. 2024) and increasing the risk of security.

A promising solution is to preserve the integrity of the cover text without any content transformations to eliminate deviation completely between stego texts (i.e., steganographic texts) and original cover texts. The core of this idea is to establish a set of steganographic coding functions to achieve a reversible mapping between a variable secret message and the same cover text, thereby eliminating the reliance on content transformations.

To this end, we propose a concept of content-preserving linguistic steganography paradigm and further present a flexible and effective LS method called CLstega (**C**ontent-preserving **L**inguistic **stega**nography) based on this paradigm. CLstega provides a practical implementation of the proposed paradigm, demonstrating a feasible path to achieving content-preserving linguistic steganography. It establishes reversible mappings between different secret messages and the same cover text by controlling probability distribution transformation through fine-tuning a masked language model (MLM), thereby enabling accurate extraction of the embedded secret message from an unmodified cover text. Specifically, CLstega utilizes an augmented masking strategy to elaborately locate and mask embedding positions to reduce the difficulty of distribution transformation. Subsequently, CLstega employs a dynamic distribution steganographic coding strategy to map secret messages to distinct

*Corresponding author: yangzl@bupt.edu.cn



(a) Content-transformation-based LS paradigm (b) Content-preserving LS paradigm

Figure 1: Frameworks of LS paradigms. Red solid arrows indicate embedding, blue dashed arrows indicate extraction.

distributions. Based on the encoding results, CLstega selects a special target word as the label for each masked embedding position, constructing a labeled and masked sentence dataset that aligns with the desired distribution. Finally, fine-tuning the MLM ensures that the original distributions are transformed into the corresponding target distributions at the embedding positions, enabling high-capacity embedding with absolute security—without modifying the cover text. The experimental results demonstrate the feasibility of the proposed content-preserving linguistic steganography method, showing that it offers the strongest security and a competitive embedding capacity compared to baselines. Our contributions are summarized as follows:

- **The first content-preserving linguistic steganography paradigm** is proposed. This paradigm provides a novel perspective to eliminate the subtle detectable deviations introduced during the embedding process. It enables perfectly secure covert communication by using unmodified natural text as the stego text.
- **A novel dynamic distribution steganographic coding method** is proposed. It establishes a mapping between the cover word and different code values by transforming its prediction distribution rather than modifying the word itself. This enables secret message embedding by associating each cover word with a specific target distribution, and supports accurate extraction by identifying the same distribution from the unmodified cover text.
- **Controllable distribution transformation** is introduced by constructing a labeled masked sentence dataset to fine-tune the pre-trained masked language model (MLM). This ensures that the target distributions align with the intended secret messages, enabling embedding while preserving the original cover text.

Background and Related Work

Content-Transformation-based Linguistic Steganography. Linguistic steganography (LS) embeds imperceptible secret messages within texts, aiming to make the resulting stego text (i.e., steganographic text) s indistinguishable from the natural cover text c, i.e., $s \approx c$, to ensure the security of secret messages. Existing methods typically achieve this through semantically approximate equivalent transformations of text content, such as synonym substitution, syntactic transformation, context-aware lexical replacement, or predicted word selection during text generation. Collectively, such methods constitute a content-transformation-based LS paradigm, where the secret message is generally embedded (or extracted) at the designated embedding position e in the cover text (or stego text). The process utilizes either a pre-defined transformation function f (e.g., synonym substitution, syntactic transformation) or a generation function g (e.g., used in various generative language models) to create a set of candidate items CW, that are approximately equivalent for the content at the embedding position e. The steganographic coding function h (e.g., Huffman coding (Dai and Cai 2019), arithmetic coding (Ziegler, Deng, and Rush 2019), fixed-length coding (Yang et al. 2018), etc.) is then applied to encode each candidate item in CW to a value in the secret message space M, establishing a surjective mapping. This ensures that there exists at least one corresponding candidate item $cw \in CW$ such that $h(cw) = m$ ($m \in M$). The overall process of the content-transformation-based LS paradigm can be formalized as:

$$\begin{cases} f/g : w_e \to CW \quad \text{(candidate set generation)} \\ h : CW \leftrightarrow M \quad \text{(steganographic coding)} \\ Emb : (w_e, m) \mapsto cw = h^{-1}(m),\ cw \in CW \\ Ext : cw \mapsto m = h(cw),\ cw \in CW \end{cases}, \quad (1)$$

where w_e denotes the original content (e.g., a word, phrase or sentence) at the embedding position e, $\to$ represents a one-to-many transformation from the original content to a set of semantically equivalent candidates CW, and $\leftrightarrow$ represents a surjective (onto) mapping between CW and the secret message space M. The embedding operation (Emb) transforms the original content w_e to a selected target candidate $cw = h^{-1}(m)$ for embedding m at position e, while the extraction operation (Ext) decodes the message m by applying h to the observed candidate cw in the stego text.

It is important to emphasize that the steganographic coding function h is a surjection, ensuring that every secret message value $m \in M$ is mapped to at least one candidate item $cw \in CW$. This implies that the candidate set CW must contain multiple items at each position e for embedding different m. Due to inherent differences in statistical, linguistic, and perceptual characteristics among candidate items, distributional shifts between stego and cover texts are inevitably

introduced, posing potential security risks.

Generally, content-transformation-based LS methods are categorized into two types: Modification-based Linguistic Steganography (MLS), which derives candidate items from existing content, and Generation-based Linguistic Steganography (GLS), which produces them anew via text generation.

Modification-based Linguistic Steganography. Initially, MLS primarily hides information by semantically equivalent replacements of existing cover text content using specific rules, such as synonym substitution (Chang and Clark 2010; Yajam, Mousavi, and Amirmazlaghani 2014; Xiang et al. 2018) and syntactic transformations (Meral et al. 2009; Chang and Clark 2012). Nevertheless, these methods are more likely to produce stego texts with syntactic unnaturalness and semantic inconsistencies. With the rapid development of pre-trained language models, recent works (Ueoka, Murawaki, and Kurohashi 2021; Zheng and Wu 2022; Xiang et al. 2023; Yang et al. 2023; Xiang, Ou, and Zeng 2023) try to leverage language models to enhance the diversity of semantically equivalent rules, thereby improving the performance of MLS. For instance, Xiang et al. (2023) combined the BERT model with classifiers, leveraging the discriminative capabilities of a CNN discriminator to construct a causal-aware network, determining suitable embedding positions based on the causal scores of the words in the original sentence, which further enhanced the security of the stego text. Yang et al. (2023) proposed a novel encoding method called "semantic-aware bins coding", utilizing translation-based paraphrasing to change the expression of a given text for embedding secret messages. Xiang, Ou, and Zeng (2023) constructed a syntactically controllable paraphrase generation model to automatically modify the syntactic attribute of the original text, thereby increasing the diversity of syntactic transformation and improving the embedding capacity. However, MLS has difficulty in improving the embedding capacity while ensuring satisfactory security due to the limited information redundancy in the text.

Generation-based Linguistic Steganography. GLS leverages text generation technology and steganographic coding algorithms to embed secret messages by controlling the selection of generated words during the automatic generation of stego texts. Recently, due to the powerful generative capacity of generative language models, GLS has made significant progress in both fluency and embedding capacity (Tina Fang, Jaggi, and Argyraki 2017; Yang et al. 2018; Dai and Cai 2019; Ziegler, Deng, and Rush 2019; Shen, Ji, and Han 2020). These methods improve the perceptual-imperceptibility of the stego text to some extent. Moreover, some advanced methods (Kaptchuk et al. 2021; Zhang et al. 2021; Ding et al. 2023b) have focused on incorporating constraint conditions to minimize the overall divergence in statistical distribution and semantic expression between cover texts and stego texts. Yang *et al.*(Yang et al. 2020b) proposed a novel VAE-stega method, which uses the encoder in VAE-Stega to learn the overall statistical distribution characteristics of a large number of normal texts, and then use the decoder in VAE-Stega to generate stego sentences that conform to both of the statistical language model and overall statistical distribution of normal sentences, thereby balancing the perceptual-imperceptibility and statistical-imperceptibility of the stego texts. Recently, with the significant progress made in large language models(LLMs), some works try to leverage the advantages of LLMs to enhance the quality and semantic richness of the generated stego text(Li et al. 2024; Bai et al. 2024; Wu et al. 2024). However, even the most advanced LM-generated text still exhibits a distribution gap compared to natural text, which brings potential security risks (Pang et al. 2024), largely due to inherent biases rooted in training data limitations (Welleck et al. 2020).

Paradigm Statement

Content-transformation-based LS methods struggle to preserve the original text distribution, as they encode secrets by altering content. This limitation prevents them from achieving perfect security. To overcome this, we propose a novel **content-preserving linguistic steganography paradigm**, which ensures the stego text is perfectly indistinguishable from the cover text by embedding messages without altering the original content. Moreover, we conducted the security analysis for two paradigms in **Appendix A**[1].

To concretely illustrate the principle of content preservation as the key to perfect security, Figure 1 show the general frameworks of two linguistic steganography paradigms. As shown in Figure 1(a), the content-transformation-based LS paradigm employs a transformation or generation function to produce a set of candidate items at the embedding position. A steganographic coding function is then used to associate these candidates with different secret message values, and message embedding is achieved by replacing or generating a candidate item accordingly. However, in practice, the selected candidate depends on the specific secret message to be embedded, meaning the content at the embedding position varies with the hidden secret message. This process may result in the selection of candidates that deviate significantly from the semantic context, introducing potential security risks. To address these issues and achieve perfect security, we propose a content-preserving linguistic steganography paradigm that eliminates content transformation by using variable coding functions, as illustrated in Figure 1(b).

Unlike the content-transformation-based LS paradigm, the content-preserving paradigm does not alter the cover text to generate the stego text. Instead, it utilizes a set of steganographic coding functions $H = \{h_0, h_1, \cdots\}$ to dynamically assign different codes to the same original text content at the embedding position e. These coding functions ensure that the same original content can be encoded/decoded into any possible value in the secret message space M, depending on which specific function is applied. For example, as illustrated in Figure 1(b), the content "discover" can be encoded/decoded into message '0' employing coding function h_0, or into a different message '1' using h_1. If the secret message to be embedded at this position is '1', the stegosystem selects coding function h_1 accordingly, while the word "discover" is preserved in the stego text. The embedding and

[1]Full appendices will be available in the extended version.

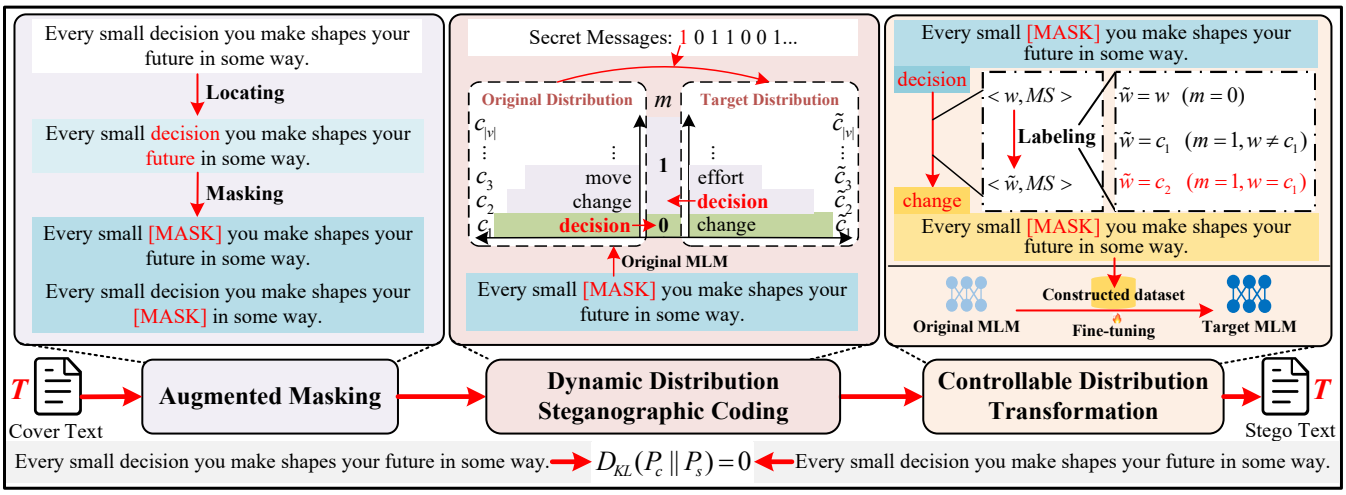


Figure 2: The overall framework of the proposed content-preserving linguistic steganography (CLstega).

extraction processes in this paradigm are defined as follows:

$$\begin{cases} H = \{h_0, h_1, \dots\} : w_e \to M \\ Emb : (w_e, m) \mapsto h_i,\ h_i \in H \\ Ext : (w_e, h_i) \mapsto m = h_i(w_e) \end{cases}, \tag{2}$$

where w_e denotes the original cover content at embedding position e, and $H = \{h_0, h_1, \dots\}$ is the set of steganographic coding functions that map w_e to different values in the secret message space M. The mapping $w_e \to M$ indicates a one-to-many relationship enabled by selecting different steganographic coding functions. During embedding, the stegosystem selects an appropriate h_i to encode w_e as m, without altering the original cover content itself. During extraction, the embedded message m is recovered by applying h_i to the unchanged w_e.

Our proposed paradigm establishes a dynamic and reversible mapping from the original cover content to the message space by selecting coding functions, supporting the embedding of arbitrary messages while preserving the content of the cover text. In **Appendix B**, we discuss the practical challenges and solution strategy of our paradigm.

Method

Building upon the proposed content-preserving LS paradigm, we present a practical linguistic steganographic method, **CLstega**, which enables secret message embedding without altering the cover text, while ensuring reliable extraction and consistency between stego and cover texts.

Overall Framework

As illustrated in Figure 2, CLstega includes three core components: *augmented masking*, *dynamic distribution steganographic coding* and *controllable distribution transformation*. Concretely, the augmented masking module locates appropriate embedding positions where the prediction distribution is easily adjustable, and constructs a masked sentence set based on these positions. Subsequently, the dynamic distribution steganographic coding module derives a target distribution by mapping the original prediction distribution at the embedding position to the encoding secret message space. Finally, the controllable distribution transformation module aligns the MLM's prediction distribution with the target distribution by fine-tuning the model on a labeled masked dataset, which is constructed by elaborately seeking appropriate target words as labels. The fine-tuned MLM is then used to encode and decode secret messages by reproducing the target prediction distribution for the original content at embedding positions, achieving embedding secret messages without modifying the cover text.

Augmented Masking

In general, given a text with certain tokens replaced by a special token, a Masked Language Model (MLM) is trained to recover the original tokens based solely on the surrounding context. To adapt our linguistic steganography task, we first determine appropriate embedding positions, replacing them with a special token to construct a masked sentence set for controllable distribution transformation.

Locating. For the given cover text T, we first segment it into sentence units using a text segmentation tool[2], where $T = \{S_1, S_2, \dots, S_L\}$, and the i-th sentence $S_i = \{w_1, w_2, \cdots, w_l\}$ consists of l words. As prediction distributions from a pre-trained MLM differ notably across lexical categories (Yang, Zhang, and Zhao 2023), we need to identify which categories of words are more conducive to precisely adjusting the prediction distribution, thereby improving the likelihood of successfully embedding and extracting secret messages.

Functional words (e.g., articles and prepositions) tend to receive low entropy predictions and are easier to predict, while non-functional words (e.g., nouns, verbs) typically exhibit higher entropy outputs, offering more flexibility for prediction distribution adjustment (Yang, Zhang, and Zhao

[2]Text segmentation tool: https://www.nltk.org/

2023). To this end, we use the POS-tagging tool[3] to identify and locate non-functional words within each sentence $S_i \in T$. The first k non-functional words in each sentence are selected as embedding positions for secret messages.

Masking. Once the embedding positions are located, a straightforward masking strategy is utilized to replace all k selected original words in S_i with the [MASK] token, resulting in a masked sentence MS_i. This strategy is referred to as Full-Position Masking (FPM), where the MLM utilizes only the remaining $l-k$ words to predict the masked tokens. However, as k increases, the loss of context may degrade the MLM's ability to predict the target words accurately.

To improve the prediction accuracy for the masked token, we introduce a Single-position augmented masking (SPAM) strategy, which creates k copies of each sentence S_i within the cover text, each containing only one [MASK] token at a distinct embedding position. Each copy (i.e., masked sentence) can retain $l-1$ tokens of context, enabling the MLM to make more context-aware predictions, thereby improving the reliability of the encoding process. As illustrated in Figure 2, when $k=2$, we locate two embedding positions in the given sentence, and then create a masked sentence for each position. Consequently, we generate two masked sentences. For a given cover text T containing L sentences, this process results in $L \times k$ masked sentences, forming the masked sentence set used in subsequent modules.

Dynamic Distribution Steganographic Coding

To keep the cover text unchanged during embedding, a feasible way is to encode the same original word into different potential secret messages by varying the prediction distribution. To this end, we propose a dynamic distribution steganographic coding (DDSC) strategy, which constructs a target prediction distribution for each embedding position based on the given secret message.

Coding rule In this work, we consider a simple coding rule for creating a one-to-one invertible mapping between codes and prediction distributions at each embedding position. Let $P = \{p_{c_1}, p_{c_2}, \ldots, p_{c_{|v|}}\}$ denote the probability distribution over the vocabulary v, sorted in descending order of predicted probability at the masked embedding position, as generated by a pre-trained MLM. Here, p_{c_j} is the predicted probability of the word c_j, the j-th ranked word in the distribution. We define the following coding rule $fr(\cdot)$:

$$fr(P) = \begin{cases} 0, & (\text{ if } p_w = p_{c_1}) \\ 1, & (\text{ if } p_w < p_{c_1}) \end{cases}, \tag{3}$$

where p_w represents the probability of the original word w under the distribution P. That is, if w has the highest predicted probability among P, the distribution P is encoded as '0', otherwise, it is encoded as '1'.

Original distribution For a masked sentence MS with a single special token [MASK], we obtain the original prediction distribution $P_o = \{p^o_{c_1}, p^o_{c_2}, \ldots, p^o_{c_{|v|}}\}$ at the masked embedding position from the pre-trained MLM, where $p^o_{c_j}$ denotes the probability assigned to word c_j at rank j. Let $C = \{c_1, c_2, \ldots, c_{|v|}\}$ denote the ranked candidate list corresponding to P_o, where c_1 represents the candidate word with the highest prediction probability.

Target distribution Denote the target prediction distribution as $P_t = \{p^t_{\tilde{c}_1}, p^t_{\tilde{c}_2}, \ldots, p^t_{\tilde{c}_{|v|}}\}$, and the corresponding ranked candidate list as $\tilde{C} = \{\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_{|v|}\}$. We divide $\tilde{C}$ into two intervals: $\tilde{C} = \tilde{C}^1 \cup \tilde{C}^2$, where $\tilde{C}^1 = \tilde{c}_1$ and $\tilde{C}^2 = \{\tilde{c}_2, \tilde{c}_3, \ldots, \tilde{c}_{|v|}\}$.

According to the coding rule (i.e., Eq. 3), the target distribution P_t must satisfy the following criterion:

$$\begin{cases} w \in \tilde{C}^1, & (\text{ if } m = 0) \\ w \in \tilde{C}^2, & (\text{ if } m = 1) \end{cases}, \tag{4}$$

where w represents the original word at the position of [MASK], and $m \in \{0, 1\}$ represent the secret message bit.

Note that if the original distribution P_o fails to satisfy the criterion, it is transformed into the target distribution P_t via the following controllable distribution transformation module to ensure successful embedding.

Controllable Distribution Transformation

To satisfy the encoding condition defined by the target distribution P_t, we must transform the original distribution P_o so that the rank of the original word w shifts from its position in the original ranked list C to the appropriate position in the target ranked list $\tilde{C}$. This transformation is achieved without modifying the cover text itself, thereby preserving content while enabling secure message embedding. To this end, we elaborately select target words as labels for masked sentences, supervising the fine-tuning of the pre-trained MLM to guide this distribution transformation. The selection of target word $\tilde{w}$ follows three cases: 1) When $m = 0$: the goal is to ensure that the original word w ranks first in the predicted list C. Thus, the target word is set as the original word itself, i.e., $\tilde{w} = w$; 2) When $m = 1$ and $w = c_1$: The goal is to displace w from the top rank so that it can be encoded as '1'. In general, a word from the second interval of C is selected as the target, typically the second-ranked word c_2, i.e., $\tilde{w} = c_2$; 3) When $m = 1$ and $w \neq c_1$: The goal is to prevent w from occupying the top position. In this case, we reinforce c_1 as the top-ranked word with the highest predicted probability, i.e., $\tilde{w} = c_1$.

All labeled pairs $\{< \tilde{w}_i, MS_i >\}$ constitute a new labeled masked dataset to fine-tune the MLM, adjusting the prediction distributions at the embedding positions to match the required target distributions. During fine-tuning, the cross-entropy loss is used to quantify the difference between the predicted probability distribution and the target word at each masked position, as follows:

$$\mathcal{L}_{ce} = -\sum\nolimits_{i=1}^{N_C} y_i \log P(w_i | w_{\setminus i}), \tag{5}$$

where N_C represents the total number of masked words, y_i is the one-hot vector corresponding to the target word, w_i is the original word at the i-th masked position, $w_{\setminus i}$ refers to the remaining context words in the i-th masked sentence,

[3]POS-tagging tool in spaCy: https://spacy.io/

and $P(w_i|w_{\setminus i})$ is the conditional probability distribution predicted by the MLM for the i-th masked position.

During secret message extraction, the receiver identifies the embedding positions in the received stego text using a shared secret key. The fine-tuned target MLM is then applied to predict the probability distributions at each embedding position. The secret message is recovered by checking whether the original word falls within the top rank or the second interval of the predicted distribution, enabling reliable secret message extraction.

Experiments and Analysis

Datasets and Implementation Details

We randomly select 10,000 English sentences from CC-100 dataset (Wenzek et al. 2020) as cover texts for the experiments. To ensure adequate embedding capacity, each sentence contains at least 10 words.

We use BERT (Kenton and Toutanova 2019), initialized with pretrained *bert-base-cased* from Hugging Face [4], as the masked language model (MLM) for masked token prediction in our experiments. For fine-tuning, we enable FP16 mixed-precision for training to improve computational efficiency. The AdamW (Loshchilov and Hutter 2019) optimizer is used with a weight decay of 0.01, an initial learning rate of 5e-5, and a batch size of 32.

Evaluation Metrics

Following previous work (Zhou et al. 2021), we use Embedding Rate (ER) to assess the embedding capacity. Accuracy (Acc) and F1 score (F1) of a steganalysis method are employed to evaluate the security of stego texts. Perplexity (PPL) is used to measure the imperceptibility of stego text. Additionally, we introduce two new metrics to evaluate the extraction performance: Extraction Success Rate (ESR) and Extraction Time (ET). A detailed description of these metrics is provided in **Appendix C**.

Baselines

To ensure a comprehensive comparison, we rebuilt the following advanced methods using their original settings.

Modification-Based Linguistic Steganography: (1) *FELS* (Ueoka, Murawaki, and Kurohashi 2021): It generates candidate words via BERT-based prediction and performs word substitutions using block coding to embed secret messages. (2) *ARLS* (Zheng and Wu 2022): It is an autoregressive LS algorithm based on BERT that utilizes consistency coding to address the limitations of block coding. (3) *CPGLS* (Xiang et al. 2023): It constructs a CNN-based causal perception network to assess the security of cover words and their BERT-predicted substitutes, ensuring controlled and secure message embedding.

Generation-Based Linguistic Steganography: (1) *ADG* (Zhang et al. 2021): It dynamically groups candidate words based on probability distribution at each time step during text generation for adaptive embedding. (2) *Discop* (Ding et al. 2023b): It embeds messages by creating multiple copies of the probability distribution, preserving the original distribution to enhance security.

[4]https://huggingface.co/google-bert/bert-base-cased

Method	BiLSTM-Dense		SeSy		HiDuNet	
	Acc↓	F1↓	Acc↓	F1↓	Acc↓	F1↓
FELS	0.6935	0.6714	0.6048	0.6245	0.6816	0.7452
ARLS	0.6420	0.6037	0.5567	0.6083	0.6352	0.6411
CPGLS	0.5130	0.5375	0.5140	0.5354	0.5390	0.5035
ADG	0.5215	0.5392	0.5534	0.5438	0.5645	0.5875
Discop	0.5085	0.5197	0.5032	0.5095	0.5082	0.5481
CLstega	**0.4955**	**0.5070**	**0.5038**	**0.4968**	**0.5012**	**0.4924**

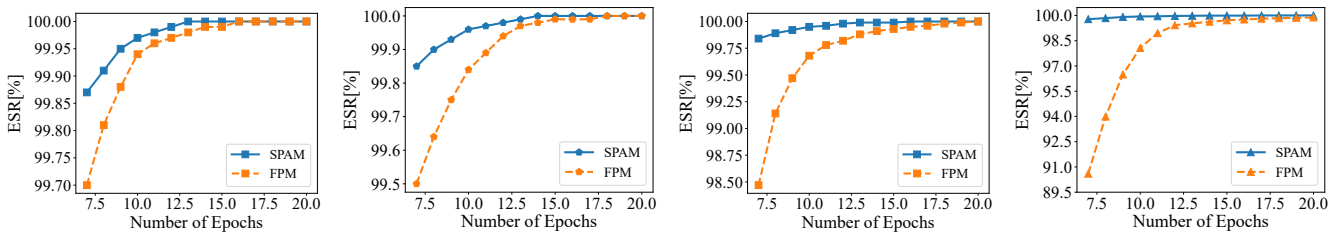
Table 1: Comparison of anti-steganalysis performance.

Results and Analysis

Extraction Success Rate Analysis. Figure 3 illustrates the extraction success rate (ESR) results of the proposed CLstega under different numbers of embedding positions k and fine-tuning epochs. $k = all$ denotes that all non-functional words are chosen as the embedding positions. We can see that ESR exhibits a consistent upward trend during fine-tuning and ultimately converges to 100%. With the number of epochs held constant, ESR generally decreases as k increases. Moreover, the proposed SPAM strategy demonstrates superior performance over FPM in the initial fine-tuning epochs for the same k. SPAM reaches 100% ESR more quickly with fewer fine-tuning epochs. Compared to FPM, SPAM creates k separate masked sentences from each original sentence, each containing only one masked position. This enables the MLM to make better use of the surrounding context when predicting a single target word. As a result, SPAM achieves higher prediction accuracy and more efficient fine-tuning, ultimately facilitating a 100% success rate for extracting secret messages from embedding positions.

Security Analysis. We select three promising linguistic steganalysis models: BiLSTM-Dense(Yang et al. 2020a), SeSy(Yang et al. 2021), and HiDuNet(Peng et al. 2023), which are designed to distinguish stego texts from cover texts. As shown in Table 1, CLstega outperforms all baselines, achieving near-random detection performance (both Accuracy and F1 score are close to 0.5) across all steganalysis models. This result confirms the perfect security of CLstega, as the steganalysis models fail to distinguish stego texts from cover texts. The core reason is that CLstega embeds secret messages without modifying the cover text, thereby eliminating any detectable artifacts that could be exploited by steganalysis models.

Imperceptibility and Embedding Capacity Analysis. As shown in Table 2, we compare the average Perplexity (PPL) of 1,000 stego texts generated by different LS methods. CLstega achieves the lowest PPL, significantly outperforming all baselines. This is because CLstega preserves the original cover text without modification during message embedding, thereby ensuring high fluency and naturalness of



(a) $k = 2$ (b) $k = 4$ (c) $k = 8$ (d) $k = all$

Figure 3: Extraction success rate for different masking strategies and numbers of embedding positions k.

Method	Parameters	PPL↓	ER↑
FELS	$f = 3, t_p = 0.02$	90.26	0.2471
ARLS	$f = 3, t_p = 0.02$	87.25	0.2542
CPGLS	$\rho = 0.02$	82.55	0.1080
ADG	$p = 1$	512.34	5.1832
Discop	$p = 1$	86.33	**5.5256**
CLstega	$k = all$	**70.16**	0.4204

Table 2: Results of imperceptibility and embedding capacity.

k	NFW	FW	AW
1	0.0378	0.0378	0.0378
2	0.0756	0.0757	0.0757
4	0.1512	0.1514	0.1514
8	0.2950	0.2960	0.2960
all	**0.4204**	**0.4199**	**0.9538**

Table 3: Embedding rates for three locating strategies.

stego texts. Furthermore, we present the case study in **Appendix D** for different methods to further illustrate the superior performance of CLstega in imperceptibility.

Compared to MLS methods (FELS, ARLS, and CPGLS), CLstega demonstrates superior embedding capacity (ER). This is because CLstega preserves the cover text entirely while embedding secret messages into its most words, achieving a significantly higher ER. Although CLstega exhibits lower embedding capacity than generation-based methods such as ADG and Discop), it far surpasses them in anti-steganalysis and imperceptibility performance, as demonstrated in Table 1 and Table 2. In addition, we further analyze the impact of embedding locating strategies on embedding capacity. The NFW, FW and AW strategies refer to selecting k indivisible non-function words, function words, and arbitrary words as embedding positions, respectively. As shown in Table 3, a larger k consistently yields higher ER across all strategies, as more embedding positions allow for embedding more secret messages. The NFW and FW strategies result in similar ER values across different k, since the proportions of functional and non-functional words in natural text are generally comparable. When using

k	NFW		FW		AW	
	FPM	SPAM	FPM	SPAM	FPM	SPAM
1	**0.0217**	**0.0217**	**0.0220**	**0.0220**	**0.0216**	**0.0216**
2	0.0220	0.0330	0.0229	0.3790	0.0218	0.0328
4	0.0226	0.0588	0.0258	0.0584	0.0223	0.0579
8	0.0255	0.1058	0.0324	0.1083	0.0257	0.1062
all	0.0302	0.1543	0.0340	0.1537	0.0827	0.3452

Table 4: Experimental results of extraction times (sec).

the AW strategy with $k = all$, CLstega achieves the highest ER of 0.9538. The deviation from the ideal ER of 1.0 arises from the exclusion of divisible words, which cannot be reliably predicted.

Extraction Efficiency Analysis. We evaluate extraction efficiency using different embedding locating strategies (NFW, FW, AW) and masking strategies (FPM, SPAM). As shown in Table 4, with Full-Position Masking (FPM), extraction time remains relatively stable regardless of k. In contrast, under Single-Position Augmented Masking (SPAM), extraction time increases with k, surpassing FPM when $k > 2$, which may be impractical for real-time applications. This is because SPAM extracts secret message iteratively, masking one embedding position at a time, resulting in a time complexity of $O(kN)$ for N number of sentences, whereas FPM processes all k embedding positions simultaneously with a lower complexity of $O(N)$. The limitations of this work are discussed in **Appendix E**.

Conclusion

We propose a novel content-preserving linguistic steganography paradigm that achieves perfect security by embedding secret messages without modifying the original cover text. Based on this paradigm, we introduce a practical and secure LS method, CLstega, which embeds secret messages through fine-tuning a masked language model to controllably adjust prediction distributions rather than altering the cover text. Experimental results demonstrate that CLstega achieves state-of-the-art security, strong extraction reliability, and high imperceptibility, validating the practical effectiveness of the proposed paradigm.

Acknowledgments

This research was supported by the Science and Technology Innovation Program of Hunan Province under Grant 2025RC3166, the National Natural Science Foundation of China (Grant No. 62302059, 62572176, U23B2023, 62472199, and 61972057), Guangdong Key Laboratory of Data Security and Privacy Preserving under Grant 2023B1212060036, the basic and Applied Basic Research Foundation of Guangdong Province (2025A1515011097), and the Outstanding Youth Project of Guangdong Basic and Applied Basic Research Foundation (2023B1515020064). This work is also supported by Engineering Research Center of Trustworthy AI, Ministry of Education.

References

Bai, M.; Yang, J.; Pang, K.; Huang, Y.; and Gao, Y. 2024. Semantic Steganography: A Framework for Robust and High-Capacity Information Hiding using Large Language Models. *arXiv preprint arXiv:2412.11043*.

Chang, C. Y.; and Clark, S. 2010. Practical linguistic steganography using contextual synonym substitution and vertex colour coding. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 1194–1203. ACL.

Chang, C. Y.; and Clark, S. 2012. The secret's in the word order: Text-to-text generation for linguistic steganography. In *Proceedings of COLING 2012*, 511–528.

Dai, F.; and Cai, Z. 2019. Towards Near-imperceptible Steganographic Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4303–4308. ACL.

Ding, C.; Fu, Z.; Yang, Z.; Yu, Q.; Li, D.; and Huang, Y. 2023a. Context-Aware Linguistic Steganography Model Based on Neural Machine Translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32: 868–878.

Ding, J.; Chen, K.; Wang, Y.; Zhao, N.; Zhang, W.; and Yu, N. 2023b. Discop: Provably Secure Steganography in Practice Based on" Distribution Copies". In *2023 IEEE Symposium on Security and Privacy (SP)*, 2238–2255. IEEE.

Hu, X.; Fu, Z.; Zhang, X.; and Chen, Y. 2023. Invisible and Steganalysis-resistant Deep Image Hiding Based on One-way Adversarial Invertible Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(7): 6128–6143.

Idres, A. A.; and Yaseen, H. I. 2023. Text Steganography Techniques: A Review. *International Research Journal of Innovations in Engineering and Technology*, 7(11): 648.

Kahn, D. 1996. The history of steganography. In *International workshop on information hiding*, 1–5. Springer.

Kaptchuk, G.; Jois, T. M.; Green, M.; and Rubin, A. D. 2021. Meteor: Cryptographically secure steganography for realistic distributions. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 1529–1548. ACM.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186. ACL.

Li, Y.; Zhang, R.; Liu, J.; and Lei, Q. 2024. A Semantic Controllable Long Text Steganography Framework Based on LLM Prompt Engineering and Knowledge Graph. *IEEE Signal Processing Letters*, 31: 2610–2614.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Mao, X.; Hu, X.; Peng, W.; Gan, Z.; Qian, Z.; Zhang, X.; and Li, S. 2024. From covert hiding to visual editing: robust generative video steganography. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2757–2765.

Meral, H. M.; Sankur, B.; Özsoy, A. S.; Güngör, T.; and Sevinç, E. 2009. Natural language watermarking via morphosyntactic alterations. *Computer Speech & Language*, 23(1): 107–125.

Pang, K.; Bai, M.; Yang, J.; Wang, H.; Jiang, M.; and Huang, Y. 2024. FREmax: A Simple Method Towards Truly Secure Generative Linguistic Steganography. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4755–4759. IEEE.

Peng, W.; Li, S.; Qian, Z.; and Zhang, X. 2023. Text steganalysis based on hierarchical supervised learning and dual attention mechanism. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31: 3513–3526.

Shen, J.; Ji, H.; and Han, J. 2020. Near-imperceptible Neural Linguistic Steganography via Self-Adjusting Arithmetic Coding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 303–313. ACL.

Simmons, G. J. 1984. The prisoners' problem and the subliminal channel. In *Advances in Cryptology: Proceedings of Crypto 83*, 51–67. Springer.

Su, W.; Ni, J.; Hu, X.; and Li, B. 2024. Efficient Audio Steganography Using Generalized Audio Intrinsic Energy With Micro-Amplitude Modification Suppression. *IEEE Transactions on Information Forensics and Security*, 19: 6559–6572.

Tina Fang, T.; Jaggi, M.; and Argyraki, K. 2017. Generating Steganographic Text with LSTMs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics-Student Research Workshop*, 100–106. ACL.

Ueoka, H.; Murawaki, Y.; and Kurohashi, S. 2021. Frustratingly Easy Edit-based Linguistic Steganography with a Masked Language Model. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5486–5492. IEEE.

Welleck, S.; Kulikov, I.; Roller, S.; Dinan, E.; Cho, K.; and Weston, J. 2020. Neural Text Generation With Unlikelihood Training. In *International Conference on Learning Representations*.

Wenzek, G.; Lachaux, M.-A.; Conneau, A.; Chaudhary, V.; Guzmán, F.; Joulin, A.; and Grave, É. 2020. CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 4003–4012. ACL.

Wu, J.; Wu, Z.; Xue, Y.; Wen, J.; and Peng, W. 2024. Generative text steganography with large language model. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 10345–10353.

Xiang, L.; Ou, C.; and Zeng, D. 2023. Linguistic Steganography: Hiding Information in Syntax Space. *IEEE Signal Processing Letters*, 31: 261–265.

Xiang, L.; Wu, W.; Xu, L.; and Yang, C. 2018. A linguistic steganography based on word indexing compression and candidate selection. *Multimedia Tools and Applications*, 77(21): 28969–28989.

Xiang, L.; Xia, J.; Liu, Y.; and Gui, Y. 2023. CPG-LS: Causal Perception Guided Linguistic Steganography. *IEEE Signal Processing Letters*, 30: 1762–1766.

Xue, Y.; Wu, J.; Ji, R.; Zhong, P.; Wen, J.; and Peng, W. 2023. Adaptive domain-invariant feature extraction for cross-domain linguistic steganalysis. *IEEE Transactions on Information Forensics and Security*, 19: 920–933.

Yajam, H. A.; Mousavi, A. S.; and Amirmazlaghani, M. 2014. A new linguistic steganography scheme based on lexical substitution. In *2014 11th International ISC Conference on Information Security and Cryptology*, 155–160. IEEE.

Yang, D.; Zhang, Z.; and Zhao, H. 2023. Learning Better Masking for Better Language Model Pre-training. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 7255–7267. ACL.

Yang, H.; Bao, Y.; Yang, Z.; Liu, S.; Huang, Y.; and Jiao, S. 2020a. Linguistic steganalysis via densely connected LSTM with feature pyramid. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, 5–10. ACM.

Yang, J.; Yang, Z.; Zhang, S.; Tu, H.; and Huang, Y. 2021. SeSy: Linguistic steganalysis framework integrating semantic and syntactic features. *IEEE Signal Processing Letters*, 29: 31–35.

Yang, T.; Wu, H.; Yi, B.; Feng, G.; and Zhang, X. 2023. Semantic-preserving linguistic steganography by pivot translation and semantic-aware bins coding. *IEEE Transactions on Dependable and Secure Computing*, 21(1): 139–152.

Yang, Z.-L.; Guo, X.-Q.; Chen, Z.-M.; Huang, Y.-F.; and Zhang, Y.-J. 2018. RNN-stega: Linguistic steganography based on recurrent neural networks. *IEEE Transactions on Information Forensics and Security*, 14(5): 1280–1295.

Yang, Z.-L.; Zhang, S.-Y.; Hu, Y.-T.; Hu, Z.-W.; and Huang, Y.-F. 2020b. VAE-Stega: linguistic steganography based on variational auto-encoder. *IEEE Transactions on Information Forensics and Security*, 16: 880–895.

Yi, B.; Wu, H.; Feng, G.; and Zhang, X. 2022. ALiSa: Acrostic linguistic steganography based on BERT and Gibbs sampling. *IEEE Signal Processing Letters*, 29: 687–691.

You, H.; Xiang, L.; Yang, C.; and Shen, X. 2024. Linguistic steganalysis via multi-task with crossing generative-natural domain. *Neurocomputing*, 603: 128260.

Zhang, R.; Liu, J.; and Zhang, R. 2024. Controllable Semantic Linguistic Steganography via Summarization Generation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4560–4564. IEEE.

Zhang, S.; Yang, Z.; Yang, J.; and Huang, Y. 2021. Provably Secure Generative Linguistic Steganography. In *Meeting of the Association for Computational Linguistics*, 3046–3055. ACL.

Zheng, X.; and Wu, H. 2022. Autoregressive linguistic steganography based on BERT and consistency coding. *Security and Communication Networks*, 2022(1): 1–11.

Zhou, X.; Peng, W.; Yang, B.; Wen, J.; Xue, Y.; and Zhong, P. 2021. Linguistic steganography based on adaptive probability distribution. *IEEE Transactions on Dependable and Secure Computing*, 19(5): 2982–2997.

Ziegler, Z.; Deng, Y.; and Rush, A. M. 2019. Neural Linguistic Steganography. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1210–1215. ACL.