

Dynamically Allocated Interval-Based Generative Linguistic Steganography with Roulette Wheel

Yihao Wang, Ruiqi Song, Lingxiao Li, Ru Zhang^{*}, Jianyi Liu

School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China.

Abstract

Existing linguistic steganography schemes often overlook the conditional probability (CP) of tokens in the candidate pool, allocating the one coding to all tokens, which results in identical selection likelihoods. This approach leads to the selection of low-CP tokens, degrading the quality of stegos and making them more detectable. This paper proposes a scheme based on the interval allocated, called DAIRstega. DAIRstega first uses a portion of the read secret to build the roulette area. Then, this scheme uses the idea of the roulette wheel and takes the CPs of tokens as the main basis for allocating the roulette area (i.e., the interval length). Thus, tokens with larger CPs are allocated more area. The secret will have an increased likelihood of selecting a token with a higher CP. During allocation, we design some allocation functions and three constraints to optimize the process. Additionally, DAIRstega supports prompt-based controllable generation of stegos. Rich experiments show that the proposed embedding way and DAIRstega perform better than the existing ways and baselines, which shows strong perceptual, statistical, and semantic concealment, as well as anti-steganalysis ability. It can also generate high-quality longer stegos, addressing the deficiencies in this task. DAIRstega is confirmed to have potential as a secure watermarking, offering insights for its development. Our codes and data are available at: <https://github.com/WangYH-BUPT/DAIRstega>.

Keywords: Linguistic steganography, Embedding way, Allocation function, Interval allocated, Roulette wheel, Large language models, Concealment

1. Introduction

Steganography, a technology of information hiding [1], embeds secrets within media such as texts [2, 3] and images, obtaining steganographic media that is perceptually indistinguishable from normal media. Only authorized receivers can perceive whether the media is steganographic and accurately extract the secrets, thereby protecting privacy. Benefiting from the ability of text to be transmitted losslessly over public channels, research on linguistic steganography has explosive growth [4, 5, 2]. To increase the anti-detection of steganographic text (stego), the focus of work has shifted from modified schemes [6] represented by synonym replacement [7] to generative schemes [8, 3].

The concealment of stego determines the success of covert communication to a great extent. Depending on the constraints, the concealment is manifested in three aspects: perceptual, statistical, and semantic. “Perceptual concealment” focuses on ensuring stegos with complement and fluency. “Statistical concealment” requires the distribution of stegos to be close to that of the natural texts (covers). “Semantic concealment” aims to stego that is coherent and semantically controllable. The concealment of stego mainly depends on the design of the steganography embedding way. An excellent embedding way ensures that the conditional probability (CP) distribution of the text is not changed as much as possible while embedding the secret. In this way, stegos are difficult to perceive and detect by unauthorized

[☆]This work is supported by the National Natural Science Foundation of China (Grant U21B2020) and supported by BUPT Excellent Ph.D. Students Foundation (Grant CX2023120, CX20241055).

^{*}Corresponding author

Email addresses: yh-wang@bupt.edu.cn (Yihao Wang), songrq123@bupt.edu.cn (Ruiqi Song), Lingxiao-Li@bupt.edu.cn (Lingxiao Li), zhangru@bupt.edu.cn (Ru Zhang^{*}), liujy@bupt.edu.cn (Jianyi Liu)

receivers. Most existing schemes have made efforts in this regard. Yang et al. [9] and Ding et al. [10] encoded each token in the candidate pool with Fixed length coding (FLC) and only used secret information to determine the next token to achieve the purpose of embedding. Yang et al. [3] and Zhou et al. [11] designed the Huffman coding-based (HC) embedding, which generates stegos with good completeness and fluency, improving perceptual concealment. Zhang et al. [12] proposed the Adaptive dynamic grouping coding-based (ADG) embedding. They achieved strong statistical concealment. Kaptchuk et al. [13] and Wang et al. [5] proposed the Arithmetic coding-based (AC) embedding. Lu et al. [4] used contextual learning and variable length coding-based (VLC) to generate stegos. It improves semantic concealment.

However, the embedding ways of almost all existing schemes allocate one or the same number of codings to the tokens in the candidate pool. That is, when embedding, the likelihood of selecting all tokens is the same. They do not consider that the CP of a portion of the tokens is not high. If the secret is determined to these tokens, the quality of stego generation will be dropped. The example is shown in the Figure 1.

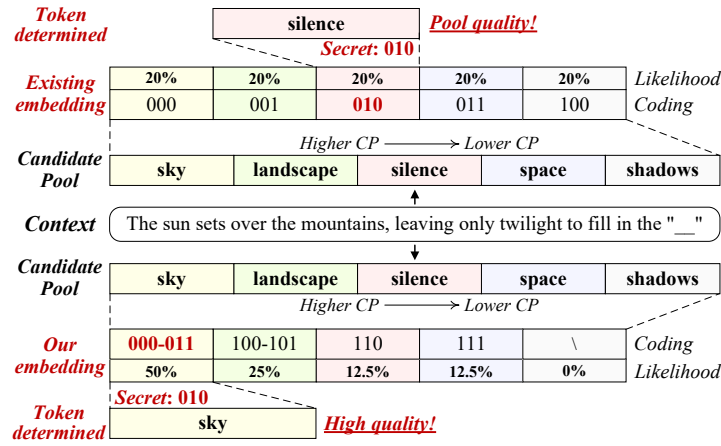


Figure 1: Examples of existing embedding ways and the proposed embedding way. After selecting n tokens with larger CPs to build a candidate pool, the existing way assigns a coding to each token in the candidate pool. Since the secret binary can be arbitrary, to embed the secret, the existing embedding way selects all tokens with the same likelihood of $1/n$. The proposed way will build a roulette area and allocate it according to the CP value to ensure that the larger the CP of the token, the larger the allocation area. Therefore, the proposed way has a higher likelihood of selecting a better token. In the end, the quality of the token determined by the proposed way is generally better than that of the existing ways.

Tokens with low CPs will reduce the quality of stego and cause significant differences in its statistical characteristics from cover. Since steganalysis detects stegos by capturing these differences [14, 15], poor-quality stego is easier to identify and has weaker anti-steganalysis capabilities. The performance of stegos obtained by different embedding ways is shown in Figure 2. Therefore, it is necessary to explore a difficult-to-perceive and detect scheme with concealed embedding.

This paper proposes the dynamically allocated interval-based steganography, called DAIRstega. Different from the existing schemes, DAIRstega introduces the idea of the roulette wheel to design the embedding way. We believe that the quality of the tokens in the candidate pool is different. Thus, the tokens need to be given different shoot likelihoods by secret, increasing the likelihood of selecting tokens with higher CP. Then, we explore the characteristics of the CP of tokens, and design a variety of relationships between CPs and the number of allocated codings. By analyzing the adaptability of these allocated functions to the invisible information-hiding tasks, the allocation expression is condensed. In addition, DAIRstega can receive instructions and directly control the stego generation.

The contributions are summarized as follows.

- To augment the concealment of stegos, DAIRstega introduces the idea of the roulette wheel, and dynamically allocates the number of codings. The functions between CPs and allocated numbers are also given. According to Occam's razor, the functions are condensed.
- To ensure the adaptability of the functions designed, we gave three constraints.

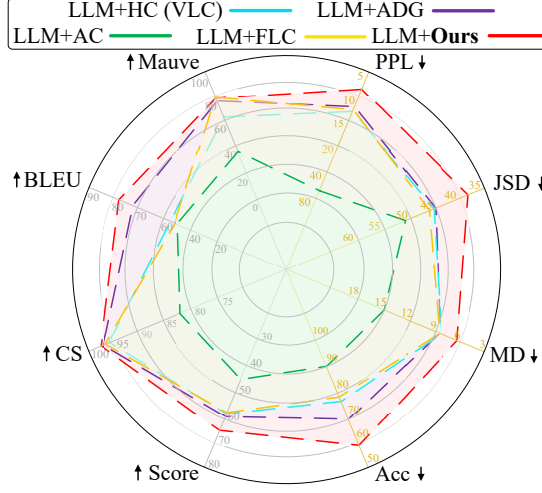


Figure 2: The result of large language model (LLM) + embedding ways. “↑” and “↓” represent the higher / lower the value, the better the result. The metrics are found in “Section 4.1”. The values are seen in Table 2.

- To make the evaluation results consistent with the LLMs era, the high-quality corpus was built and used this corpus as a cover to train the model of all schemes.
- To verify the effectiveness of the proposed embedding way and DAIRstega, we performed experiments regarding perceptual, statistical, and semantic concealment, as well as anti-steganalysis performance.
- Experiment verified that DAIRstega can also be regarded as a secure watermarking scheme with high quality, consistency, and unforgeability.

2. Related work

The concealment of stego determines the success of covert communication to a great extent. The concealment is primarily manifested in three aspects: perceptual, statistical, and semantic.

“Perceptual concealment” focuses on ensuring that the scheme generates stego with complete and fluent. This is the most basic requirement of steganography. In pursuit of this objective, Fang et al. [16] proposed an LSTM-based steganography scheme. This scheme segments the vocabulary into several sets based on bit blocks. It then selects tokens with the highest probability that corresponds to the secret information from the candidate pool. Ding et al. [10] combined the conditional generation strategy with the replacement technique, using text sequences as auxiliary data in the stego-generation process to enhance the embedding capabilities. Yang et al. [9] trained a language model to generate stegos. Using HC-based (Huffman coding) and FLC-based (Fixed length coding) to encode the candidate pool, these stegos have excellent completeness and fluentness.

“Statistical concealment” requires the stegos distribution to be close to the covers. To achieve this goal, Yang et al. [3] designed an encoder-decoder structure, and VLC-based (Variable length coding) is used to encode the CP. To further reduce the distribution difference between cover and stego, Zhang et al. [12] used adaptive dynamic grouping coding to embed the secret recursively. Zhou et al. [11] used a generative adversarial network (GAN) to design an adaptive probability distribution steganography. These ensure statistical concealment.

“Semantic concealment” aims to generate a stego that is coherent and semantically controllable. To this end, Li et al. [8] put forward a steganography scheme based on the knowledge graph. Yang et al. [2] utilized semantic information encoding to embed secret information, realizing the effect of maintaining semantics and increasing the embedding capacity during the translation process. Wang et al. [5] leveraged the relevance of social network context to enhance contextual semantic relevance while maintaining embedding rates. Lu et al. [4] used contextual learning and GPT2 guidance to generate stegos.

3. Methodology

3.1. Modeling

For more covert communications, Alice needs to ensure that the discourse (the style of humans and the genre of texts) is the same as the covers, as shown in Figure 3.

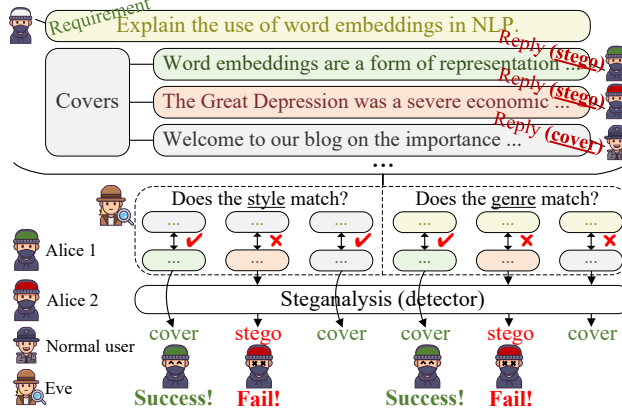


Figure 3: Example of texts being transmitted and detected. Responses that match the discourse are difficult to perceive, thus reducing the risk of steganalysis, such as Alice 1.

The covert communication hopes to avoid Eve’s perception and detection and needs to satisfy:

$$\begin{cases} d(P_C, P_S) \leq \varepsilon \\ C \Rightarrow C_{style} \sim P_{C1}, S_k \Rightarrow S_{style} \sim P_{S1} \\ C \Rightarrow C_{genre} \sim P_{C2}, S_k \Rightarrow S_{genre} \sim P_{S2} \\ (d(P_{C1}, P_{S1}) \leq \varepsilon_1) \wedge (d(P_{C2}, P_{S2}) \leq \varepsilon_2) \end{cases}, \quad (1)$$

where, P_C and P_S are the distribution of cover and stego, $d(\cdot)$ is the difference, ε is a very small value greater than 0. \sim is following a certain distribution. The above assumptions cover most situations where the secret is transmitted. Complying with the Formula 1 can improve the concealment and ensure the conduct of covert communication. The results of this aspect are shown in Table 5.

3.2. DAIRstega Overall

This scheme can generate stegos that are strongly concealed and controllable. It can satisfy the extension of covert communication, i.e., Formula 1. To make the generated stegos more consistent with the CP distribution of covers, DAIRstega uses the idea of the roulette wheel to allocate tokens dynamically in the candidate pool to coding intervals of different lengths. That is, the larger the CP is, the larger the roulette area will be allocated, and the greater the shoot that this token will be selected by secret. Instructions can be public topics, posted questions, or other requirements. Figure 4 shows the DAIRstega’s framework.

3.3. Details

Stego generation process. It is composed of the “CP (conditional probability) generation” (CPG) and “Embedding” (Em) modules. The process formulas are as follows:

$$\begin{cases} CPG : LM(I) \Rightarrow \mathbf{P} \\ Em : \text{locate } \{m_t \text{ in } DAI(BC(\mathbf{P}))\} \Rightarrow i, \\ \quad \text{match } \{i(R_{begin}) \text{ and } i(R_{end})\} \\ i = Em(CPG(I)) \end{cases}, \quad (2)$$

where, I and \mathbf{P} are the index sequence of a given text and the CP distribution of the next token, $\mathbf{P} = [p_1, \dots, p_v]$, p_v is the CP of the v -th token in the vocab. “ $LM(\cdot)$ ” is the CP calculation process. “ $DAI(\cdot)$ ” is the dynamically allocated

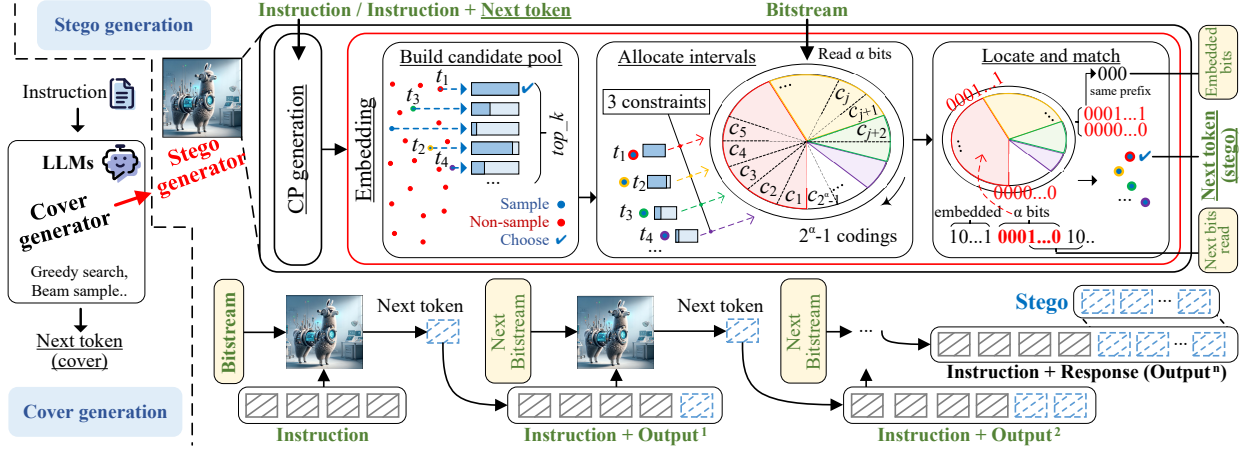


Figure 4: The DAIRstega’s framework. The “stego generation process” consists of two parts: the “CP (conditional probability) generation” and “Embedding” modules. Different from existing works, DAIRstega uses the idea of the non-uniform roulette wheel to dynamically allocate different roulette areas (numbers of codings) to tokens in the candidate pool. In this way, the secret is more likely to fall on the tokens with larger CPs, ensuring the quality of stegos. In addition, DAIRstega can not only receive secret information (in the form of bitstreams), but also input the instruction, and finally generate stego that conforms to the instruction.

interval process. “ $BC(\cdot)$ ” is the build candidate pool process. “ $\text{locate } \{a \text{ in } \circ\} \Rightarrow i$ ” means that i is the next token determined by a . “ $\text{match } \{i(R_{begin}) \text{ and } i(R_{end})\}$ ” is the same prefix that matches the beginning and end of i ’s range.

CP generation module. The input is the instruction or the instruction with part of the subsequent token generated, and the output is the CP distribution \mathbf{P} of the next token.

Embedding module. To ensure that the embedding does not excessively affect the stego quality. In our work, low-CP tokens should not be added to the candidate pool. We sort and intercept the first top_k tokens with higher CPs to build the candidate pool. The \mathbf{P} normalization $\mathbf{P}' = [p'_1, \dots, p'_{top_k}]$ of tokens in candidate pool is:

$$p'_j = \frac{p_j}{\sum_{j=1}^{top_k} p_j}, p'_j \in \mathbf{P}', p_j \in \mathbf{P}. \quad (3)$$

Then, the α -bit secret bitstream is read to determine the total number $(2^\alpha - 1)$ of intervals. The different numbers of consecutive codings are allocated to tokens in the candidate pool. Intuitively, the allocated relationship between \mathbf{P}' and the number of allocated codings $\mathbf{N} = [n_1, \dots, n_{top_k}]$ can be linear, the linear function is:

$$n_j = \lfloor \beta p'_j \times (2^\alpha - 1) \rfloor, n_j \in \mathbf{N}, \quad (4)$$

where, β is the coefficient, $\lfloor \cdot \rfloor$ is the rounding operation. After encoding, the number of n_j is proportional to p'_j . It also ensures that the likelihood of selecting a larger p'_j token is greater, directly ensuring the stego quality. To accurately recover the secret, the bits embedded are not the α bits at this moment, but the same prefix at the beginning and end of this interval. The next α -bit secret read starts with the next bit embedded this time.

Further exploration and discovery, if the linear function is used, the payload (unit: bit per word) can be improved. Tokens with very high p will be allocated to most codings, and a large interval length will result in a short prefix of this interval regardless of how many bits are read at a time. Longer stego needs to be generated to embed the given bits, resulting in a lower payload. So, we try to design the nonlinear functions to improve the payload. Using logarithmic and exponential operations, a variety of nonlinear allocations can be designed. However, not all nonlinear allocations are suitable for information hiding. We define three constraints that the functions need to meet. x is a variable that meets all CPs. Since the CP value of the token in the candidate pool is relatively large, ϵ^+ is a smaller value greater than 0, such as 0.1, 0.2, and so on. Although the CP distribution is discrete, the independent variable of the distribution function is continuous, so it is expressed here using derivatives.

Constraint 1: (*Guaranteed text diversity*)

$$DAI(x) \geq x, \forall x \in [\epsilon^+, 1). \quad (5)$$

If the steganographic embedding can only select a token with extremely high CP, the generated diversity will be lost, and the existence of the stegos will be easily exposed. This constraint indicates that the number of codings allocated to tokens with lower CPs should be appropriately increased to enhance their selection likelihood.

Constraint 2: (*Guaranteed text quality*)

$$\frac{\partial DAI(x)}{\partial x} \geq \epsilon^+, \forall x \in [\epsilon^+, 1), \quad (6)$$

The existing embedding ways assign the same number of codings to all tokens, which results in the same likelihood of randomly selecting each token, thus affecting the generation quality. Therefore, this constraint requires that tokens with larger CP need to be assigned a longer interval, that is, the number of allocations needs to increase with the increase of CP, and the change rate of the number of allocations cannot be too small. If $\frac{\partial DAI(x)}{\partial x} \leq \epsilon^+$, which means the number of allocation tokens is similar. It will degenerate into a similar allocation of the same number of codings as the existing work, thereby increasing the likelihood of selecting poor tokens.

Constraint 3: (*Guaranteed to facilitate embedding*)

$$\frac{\partial DAI(x)}{\partial x} \leq \frac{\partial x}{\partial x} \cap \frac{\partial^2 DAI(x)}{\partial x^2} \leq 0, \forall x \in [\epsilon^+, 1). \quad (7)$$

Steganography requires an algorithm that is conducive to embedding information. If you just want high-quality tokens to be selected and blindly increase the interval length of tokens with high CPs, the same prefix at the beginning and end of the interval will be shortened or even non-existent. This is not conducive to embedding secrets and deviates from the basic function of steganography. Therefore, in order to ensure the payload, it is necessary to avoid high CP tokens occupying too much interval length, and the growth trend of $DAI(x)$ should be gradually slow, that is, mathematically, it can be achieved by limiting the second-order derivative to be less than 0. When the conditions are equal, the corresponding function will be transformed into a linear function.

Here, we give several nonlinear functions:

$$n_j = \begin{cases} \lfloor \sqrt[3]{p_j'} \times (2^\alpha - 1) \rfloor \\ \lfloor (1 - e^{-2p_j'}) \times (2^\alpha - 1) \rfloor \\ \lfloor \frac{\log_2(p_j') + b}{b} \times (2^\alpha - 1) \rfloor, b \geq 2 \end{cases} \quad (8)$$

According to the principle of Occam's razor, if you can simplify, you can do less complicated things. We condense the linear and nonlinear functions into a "simple expression" of dynamically allocated intervals that satisfy the above three constraints, as shown below:

$$n_j = \lfloor (\frac{p_j'}{\sum p'})^\beta \times (2^\alpha - 1) \rfloor, \beta \in (0, 1), \alpha \in \mathbb{Z}^+. \quad (9)$$

The smaller β is, the smaller the interval of tokens with larger CP will be, while still satisfying the premise that the interval of tokens with larger CP is larger. It will increase the length of the prefix, while reducing the length of stegos to be generated, improving the payload.

Please note that these functions are not fixed, here are just a few examples that meet the constraints. These constraints are not exhaustive, and the final condensed expression is also for simplicity and easy to understand. This can just provide thinking for subsequent research that is more conducive to hiding information.

Loop the stego generation, and stego will be obtained. Algorithm 1 illustrates the embedding process.

Secret extraction process. Bitstream extraction and embedding are a pair of inverse operations. Same as Discop [17], we also need to use the same scheme, initial context (instruction), and stegos. The instructions remain requirements visible to everyone in social networks. The formula of the extraction process Ex is as follows:

Algorithm 1 Secret embedding of DAIRstega.

Input: $B = \{x_0, x_1, \dots, x_m\}, x \in \{0, 1\}$; Instruction.

Output: Stego $S = [t_{n+1}, t_{n+2}, \dots, < \text{EOS} >]$.

```
1: while Not the end of the bitstream do
2:   Map Instruction  $\Rightarrow I_1 = [i_1, i_2, \dots, i_n]$ ;
3:   while Not  $I_{<\text{EOS}>}$  in  $I$  do
4:     Get  $\mathbf{P}$  of  $i_k, k \in [n+1, n+2, \dots, <\text{EOS}>]$ ;
5:     Build candidate pool and get  $\mathbf{P}'$  by Formula 3;
6:     Read the unembedded  $\alpha$ -bit bitstream;
7:     Give  $\beta \Rightarrow n_j$  by Formula 9;
8:     Allocate  $\mathbf{N}$  to  $I$ ;
9:     Determine the interval by the  $\alpha$  bits;
10:  end while
11:  Locate  $\{\alpha\text{-bit } B \text{ in interval}\} \Rightarrow i_k$ ;
12:  Match  $\{i_k(R_{begin}) \text{ and } i_k(R_{end})\} \Rightarrow \text{embedded secret at this moment}$ ;
13: end while
14: Map  $[i_{n+1}, \dots, i_{<\text{EOS}>}] \Rightarrow [t_{n+1}, \dots, <\text{EOS}>]$ .
```

$$\begin{cases} Ex : \text{locate } \{i_{n+1} \text{ in } DAI(BC(P))\}, \\ \quad \text{match } \{i(R_{begin}) \text{ and } i(R_{end})\} \\ B_j = Ex(CPG(I)), B_j \in B \end{cases} . \quad (10)$$

The interval is located by the stego's next token at this moment, and the same prefix is the currently extracted secret.

4. Experiments

To ensure fairness and reliability in comparisons between schemes, each result was repeated 5 times and averaged to provide the results. All experiments are run on the Linux Server with 24G NVIDIA GeForce RTX 4090 GPUs.

4.1. Settings

Dataset. We obtained a large amount of data from Wikipedia, Twitter, GPT4 [18], and the publicly available data for fine-tuning LLMs¹. The final dataset contains 57,414 texts and nearly 100 different discourse characteristics to fine-tune the language models of all schemes.

Model configuration. LLM is divided into closed-source and open-source LLMs, closed source such as Gemini, GPT4, etc., open-source such as LLaMA2-7B, etc. For detailed information on closed-source LLMs, please refer to the review of comprehensive research in related fields [19, 20]. Since DAIRstega needs to modify the token sampling inside LLMs, this paper uses open-source LLMs for experiments. DAIRstega can access any open-source LLMs. In the comparison experiments, the LLaMA2-7B model is used [21] as the base LLM. The LLaMA2-7B and LLaMA2-13B models are used in the ablation experiments of the different LLMs, and all models are operated with 16-bit precision. For simplicity, we use $\beta = 1, 0.5$ and $\alpha = 8, 32, 48$ to perform DAIRstega.

Baselines. Since the steganography scheme consists of a language model and an embedding way, the language models of existing steganography are different. Therefore, to ensure fairness in the comparison, we not only compared the schemes, but also compared the performance of different embedding ways. That is to say, comparison is divided into two parts: one is the comparison between the proposed embedding way and the existing embedding ways, and the other is the comparison between DAIRstega and the prevalent works in recent years.

¹ <https://github.com/tloen/alpaca-lora/tree/main>

· In terms of the embedding ways, the existing works can be summarized as follows: (1) Huffman coding (HC) (or Variable length coding (VLC)), (2) Adaptive dynamic grouping (ADG), (3) Arithmetic coding (AC), and (4) Fixed length coding (FLC). (1) **HC** is used in VAE-Stega [3], LSCS [4], RNNstega [9], GAN-APD [11], JLS [10], Discop [17]. (2) **ADG** is used in Hi-Stega [5] and ADG [12]. (3) **AC** is used in VAE-Stega [3], Hi-Stega [5], and Meteor [13]. (4) **FLC** is used in RNNstega [9], JLS [10].

· In terms of the prevalent works, we selected the steganography and watermarking works that showed excellent performance as the baselines. **Steganography-task baselines:** (1) **Fang** [16]. (2) **RNNstega** [9]. (3) **ADG** [12]. (4) **LSCS** [4]. **Related-task baseline:** (5) **PLMmark** [22]. It uses pre-trained language models to provide the first secure black-box watermarking scheme. They can compare the generation effects of DAIRstega from different aspects, and these performances have been widely recognized.

Evaluation metrics. ↓ means the lower the value, the better, ↑ means the higher the value, the better.

· In terms of perceptual concealment (text quality), we adopt (1) **PPL** ↓ (stego perplexity). [23, 10]. (2) **ΔPcs** ↓ (The difference between PPL of cover and stego [3]).

· In terms of statistical concealment (statistical analysis), we adopt (3) **CS** ↑. (Cosine similarity). (4) **JSD** ↓, which averages the KL divergence. (5) **ED** ↓ (Euclidean distance). (6) **MD** ↓ (Manhattan distance). (7) **ADP** ↓ (Dot product difference [24]).

· In terms of semantic concealment (discourse matching), we adopt (8) **ΔLDA** ↓. (9) **Mauve** ↑ [25]. Mauve is a metric that considers fluency and consistency. (10) **BLEU** ↑ (Bilingual evaluation understudy [26]). (11) **Score (BERTScore)** ↑ [27]. Score considers the similarity at the segment level. ·

In terms of anti-steganalysis, steganalysis is essentially a detection method [28]. It can detect whether there is secret information in the text [29]. we use the high-performance linguistic steganalysis LS_CNN [30], TS_CSW [31], EILG [32], and UP4LS [33]: (12) **Acc** ↓ (Accuracy) and (13) **F1** ↓ (F1-score).

Experimental Overview. We provide an overview of each part of the experiments, as shown in Table 1.

Table 1: Overview of each part of the experiments. It includes: experimental part (Focus), Section, Metrics, Base LLM, Table, and supplementary materials for table (Supplement).

Focus	Section	Metrics	Base LLM	Table	Supplement
Embedding ways	4.2	Perceptual, Statistical, Semantic, Anti-steganalysis		2	10, 11
Comparison study	4.3.1	Perceptual	LLaMA2-7B	3	–
	4.3.2	Statistical		4	–
	4.3.3	Semantic		5	12
	4.3.4	Anti-steganalysis		6	13, 14
Related-task baseline	4.4	Quality, Consistency, Unforgeability		7	–
DAIRstega exploration	4.5	Perceptual, Semantic	LLaMA2-7B	8	15
	4.6	Perceptual, Statistical, Semantic, Anti-steganalysis	LLaMA2-7B, 13B	9	–

4.2. Comparison with the different embedding ways

To illustrate the effect of the embedding way, we performed this experiment. We add the existing works with the same LLMs as ours. The results are shown in Table 2.

Table 2: The overall results of different embedding ways. The embedding rate of each way is near 1Bpw (bit per word). **Bold** and “*” are the best and the suboptimal result. The “LLM”s in this table are all LLaMA2-7B. For the steganography works involved in these embedding ways, see “Section 4.1”. “↑” is that the higher the value, the better the result. “↓” means the lower the value, the better the result. The complete data can be seen in Table 10 and Table 11 in Appendix 7.1.

Embedding ways	Perceptual		Statistical					Semantic \leftrightarrow	
	PPL ↓	Δ Pcs ↓	CS ↑	JSD ↓	ED ↓	MD ↓	Δ DP ↓	Mauve ↑	BLEU ↑
LLM+HC	13.331	5.639	96.54	45.79	0.263	8.255*	3.46	70.60	54.26
LLM+ADG	<u>12.750*</u>	<u>5.058*</u>	<u>97.20*</u>	<u>45.59*</u>	<u>0.240*</u>	8.316	<u>2.80*</u>	77.86	<u>71.79*</u>
LLM+AC	73.454	65.762	82.16	51.64	0.746	15.415	17.84	35.64	50.92
LLM+FLC	12.883	5.190	97.04	46.56	0.241	8.280	2.96	79.48	54.88
LLM+Ours	8.030	0.342	97.43	38.60	0.227	6.341	2.57	<u>78.53*</u>	77.42

Embedding ways	\hookrightarrow Score ↑	Anti-steganalysis							
		LS_CNN [30]		TS_CSW [31]		EILG [32]		UP4LS [33]	
		Acc ↓	F1 ↓	Acc ↓	F1 ↓	Acc ↓	F1 ↓	Acc ↓	F1 ↓
LLM+HC	58.44	77.62	78.41	78.09	78.25	78.58	78.29	86.73	86.13
LLM+ADG	<u>59.18*</u>	<u>70.10*</u>	<u>70.17*</u>	<u>69.45*</u>	<u>69.34*</u>	<u>70.99*</u>	<u>70.49*</u>	<u>84.00*</u>	<u>84.23*</u>
LLM+AC	45.10	90.15	89.98	89.69	89.37	86.92	86.52	93.62	93.37
LLM+FLC	58.44	78.53	80.05	82.72	83.32	76.67	76.15	89.03	89.20
LLM+Ours	65.41	58.95	58.74	55.28	61.69	65.01	65.83	72.92	73.30

1. HC (Huffman coding) is used in VAE-Stega [3], LSCS [4], RNNstega [9], GAN-APD [11], JLS [10], and Discop [17].

2. ADG (Adaptive dynamic grouping) is used in Hi-Stega [5] and ADG [12].

3. AC (Arithmetic coding) is used in VAE-Stega [3], Hi-Stega [5], and Meteor [13].

4. FLC (Fixed length coding) is used in RNNstega [9] and JLS [10].

Table 2 shows that the proposed embedding way can improve the stego quality (perceptual concealment), better simulate the CP distribution of covers (statistical concealment), and get the optimal anti-steganalysis. Overall performance performs better than the existing ways.

4.3. Comparison with steganography-task baselines

4.3.1. Perceptual concealment (text quality)

Table 3 shows the comparison regarding the perceptual concealment.

Table 3: Perceptual concealment comparison of stegos generated by DAIRstega and steganography baselines. **Bold** is the best result. “*” is the suboptimal result. α and β can be found in Formula 9. The meanings of the parameters “bin”, “bit”, and “ τ ” are the same in Fang, RNNstega, and ADG schemes.

Schemes	Param / Bpw	PPL ↓	Δ Pcs ↓
Fang [16]	bin = 1 / 1.00	235.271	227.578
	bin = 3 / 3.00	269.303	261.611
RNNstega [9]	bit = 1 / 1.00	79.987	72.294
	bit = 3 / 2.63	99.302	91.610
ADG [12]	$\tau = 0.5 / 4.38$	197.498	189.806
LSCS [4]	- / 1.12	9.248	1.556
DAIRstega	$\alpha = 8 / 1.10$	7.686	0.006
	$\beta = 1$ $\alpha = 32 / 1.11$	<u>8.184*</u>	<u>0.492*</u>
	$\alpha = 48 / 1.13$	8.219	0.527
	$\alpha = 8 / 1.89$	<u>13.775</u>	<u>6.082</u>
	$\beta = 0.5$ $\alpha = 32 / 2.56$	20.876	13.184
	$\alpha = 48 / 2.58$	22.069	14.377

According to the results in Table 3, it can be found that the stego quality of DAIRstega markedly surpasses that

of the baselines. This shows that DAIRstega can be more consistent with the linguistic characteristics of covers, enhancing the perceptual concealment of the stegos.

4.3.2. Statistical concealment (statistical analysis)

Table 4 shows the comparison between DAIRstega and baselines regarding the statistical analysis between covers and stegos.

Table 4: Statistical concealment comparison of stegos generated by DAIRstega and steganography baselines. The meanings of “**Bold**”, “*”, “bin”, “bit”, and “ τ ” are the same in Table 3.

Schemes	Param		CS \uparrow	JSD \downarrow	ED \downarrow	MD \downarrow	Δ DP \downarrow
Fang [16]	bin = 1		9.31	63.43	1.347	17.911	90.69
	bin = 3		9.30	63.34	1.347	17.876	90.70
RNNstega [9]	bit = 1		95.97	41.51	0.284	7.530	4.03
	bit = 3		96.32	39.38	0.271	7.371	3.68
ADG [12]	$\tau = 0.5$		96.82	43.02	0.252	8.232	3.18
LSCS [4]	-		94.20	40.28	0.340	7.182	5.80
DAIRstega	$\alpha=8$		97.20	38.86	0.237	6.458	2.80
	$\beta=1$ $\alpha=32$		97.65	38.48	0.217	<u>6.293*</u>	2.35
	$\alpha=48$		<u>97.45*</u>	38.46	<u>0.226*</u>	6.273	<u>2.55*</u>
	$\alpha=8$		<u>97.39</u>	37.80	0.229	7.204	2.61
	$\beta=0.5$ $\alpha=32$		92.43	39.69	0.389	8.976	7.57
	$\alpha=48$		92.44	<u>38.31*</u>	0.389	8.592	7.56

According to the comparison results in Table 4, it can be found that DAIRstega can better simulate the statistical distribution of cover, enhancing the statistical concealment.

4.3.3. Semantic concealment (discourse matching)

Table 5 shows the comparison regarding semantic concealment.

Table 5: Semantic concealment comparison of stegos generated by DAIRstega and steganography baselines. The unit is %. The complete data can be seen as Table 12 in Appendix 7.2.

Schemes	Param		Δ LDA \downarrow	Mauve \uparrow	BLEU \uparrow	Score \uparrow
Fang [16]	bin = 1		11.964	2.15	0.90	42.58
	bin = 3		11.964	1.96	0.90	41.65
RNNstega [9]	bit = 1		0.062	12.73	1.87	52.57
	bit = 3		18.919	11.97	1.83	52.36
ADG [12]	$\tau = 0.5$		15.901	18.54	2.72	47.55
LSCS [4]	-		0.022	2.40	31.95	59.22
DAIRstega	$\alpha=8$		<u>0.017*</u>	<u>79.91*</u>	76.37	65.76
	$\beta=1$ $\alpha=32$		0.022	76.97	<u>77.85*</u>	65.22
	$\alpha=48$		0.020	78.70	76.53	<u>65.25*</u>
	$\alpha=8$		0.018	80.83	73.19	<u>62.49</u>
	$\beta=0.5$ $\alpha=32$		0.029	75.58	66.99	60.89
	$\alpha=48$		0.006	72.40	79.64	60.29

4.3.4. Anti-steganalysis

Table 6 shows the anti-steganalysis comparison of DAIRstega and baselines.

Table 6: Anti-steganalysis (Acc \downarrow) comparison of stegos generated by DAIRstega and steganography baselines. The unit is %. The complete data can be seen as Table 13 in Appendix 7.2. We also give the detection of the other 7 steganalysis works [14, 34, 35, 36, 37, 38, 15] (including non-BERT, BERT, and LLM-based). It can be seen as Table 14 in Appendix 7.3.

Schemes	Param		LS_CNN [30]	TS_CSW [31]	EILG [32]	UP4LS [33]
Fang [16]	bin = 1		99.70	88.34	99.50	99.75
	bin = 3		99.75	89.03	99.52	99.88
RNNstega [9]	bit = 1		85.81	81.39	85.86	97.42
	bit = 3		82.93	80.35	82.71	96.78
ADG [12]	$\tau = 0.5$		85.86	82.23	84.86	98.34
LSCS [4]	-		78.86	81.19	84.15	96.22
DAIRstega	$\alpha=8$		59.50	56.82	67.25	75.40
	$\beta=1, \alpha=32$		<u>59.35*</u>	54.05	62.28	71.97
	$\alpha=48$		58.00	<u>54.98*</u>	<u>65.51*</u>	<u>71.40*</u>
	$\alpha=8$		62.88	<u>63.13</u>	<u>65.67</u>	69.04
	$\beta=0.5, \alpha=32$		72.06	68.73	71.71	76.41
	$\alpha=48$		70.17	69.48	70.64	79.46

4.4. Comparison with related-task baseline

The rise of LLMs has spawned many excellent watermarking papers [22, 39]. They provide solutions for copyright protection in terms of high text quality [39], consistency [40], and unforgeability [22]. Excitingly, DAIRstega can also be regarded as an invisible watermarking with high quality, consistency, and unforgeability.

The focus of this paper is not watermarking, we only compare it with the PLMmark, as shown in Table 7.

Table 7: Quality, consistency, and unforgeability comparison with watermarking baseline. The complete data can be seen in Table 12 and Table 13 in Appendix 7.2.

Scheme	Quality		Consistency			Unforgeability		
	PPL \downarrow	Δ Pcs \downarrow	Δ LDA \downarrow	Mauve \uparrow	Score \uparrow	TS_CSW \downarrow	EILG \downarrow	UP4LS \downarrow
PLMmark [22]	18.748	11.056	0.009	7.22	43.50	98.81	99.92	99.85
$\beta=1, \alpha=8$	7.686	0.006	<u>0.017*</u>	<u>79.91*</u>	65.76	56.82	67.25	75.40
$\beta=1, \alpha=32$	<u>8.184*</u>	<u>0.492*</u>	0.022	76.97	65.22	54.05	62.28	71.97
$\beta=1, \alpha=48$	8.219	0.527	0.020	78.70	<u>65.25*</u>	<u>54.98*</u>	<u>65.51*</u>	<u>71.40*</u>
$\beta=0.5, \alpha=8$	13.775	6.082	0.018	80.83	62.49	63.13	65.67	69.04
$\beta=0.5, \alpha=32$	20.876	13.184	0.029	75.58	60.89	68.73	71.71	76.41
$\beta=0.5, \alpha=48$	22.069	14.377	0.006	72.40	60.29	69.48	70.64	79.46

According to the comparison results in Table 7, the watermark texts generated by DAIR have excellent quality and consistency, and it is difficult for unauthorized receivers to detect the existence of the watermark text, avoiding the problem of watermark abuse. Only authorized receivers with the same model settings can detect and fully extract the watermark information.

4.5. Longer stegos by DAIRstega

Table 8: Examples of long stegos generated by DAIRstega. Bitstream is the binary of the secret. Secret can be not only texts, but also other modalities. In addition, we embed the image information. For details, see Table 15 in Appendix 7.4.

Secret&Bitsream	Instruction	Stegos	PPL	Δ LDA
Secret: Love and peace Bitstream: 010011000110 111101110110 011001010010 000001100*00 101101110011 001000010000 001110000011 00101*011000 010110001101 100101 (114bit)	Discuss the causes of the Great Depression.	The Great Depression was a severe economic downturn that lasted from 1929 to 1939. It was the worst economic depression in the history of the United States and it affected many countries around the world. The causes of the Great Depression are numerous and complex, but some of the main factors include overproduction, the bursting of the stock market bubble, banking failures, and government policies. One of the primary causes of the Great Depression was overproduction. The United States economy was booming in the 1920s, and many businesses and industries expanded their production to meet the growing demand. However, this resulted in an oversupply of goods and services, which led to a decrease in prices and a decrease in demand. This decrease in demand caused a drop in profits, which led to businesses cutting back on production and laying off workers.	2.145	0.004
	Explain the use of word embeddings in Natural Language Processing.	Word embeddings are a form of representation learning in natural language processing (NLP). Word embeddings are vectors of real numbers that represent the meaning of words. These vectors are learned from large amounts of text data and can be used to model the relationship between words and their contexts. Word embeddings can be used in various NLP tasks such as sentiment analysis, machine translation, and text classification.	3.399	0.006
	Give some strategies for getting over a fear of public speaking.	Public speaking is a common fear for many people, and it can be difficult to overcome. Here are some strategies that can help: 1. Practice: The more you practice public speaking, the more comfortable you will become. Start by practicing in front of a small group of friends or family members, and gradually work your way up to larger audiences. 2. Visualize: Before giving a speech, visualize yourself giving a successful presentation. Imagine yourself confidently delivering your speech, and see yourself receiving positive feedback from the audience. 3. Breathe: Take deep, slow breaths before giving a speech.	2.046	0.007

The secret will change the next token. This makes it difficult to maintain logical coherence when generating longer stegos. So, longer stegos with high quality is a problem, and there is little research in this task. Fortunately, we found that DAIRstega also has this ability. Examples of embedded text and cross-modal secrets are shown in Table 8 and Table 15.

4.6. Ablation of the different LLMs

We performed ablation experiments on different LLMs, and the results are shown in Table 9.

Table 9: Ablation on different LLMs. **Bold** is the best result. Here we only take “ $\beta = 1, \alpha = 32$ ” as an example.

$\beta=1, \alpha=32$	Perceptual		Statistical			
	PPL ↓	CS ↑	JSD ↓	ED ↓	MD ↓	Δ DP ↓
LLaMA2-7B	8.184	97.65	38.48	0.217	6.293	2.35
LLaMA2-13B	8.431	96.32	43.93	0.271	8.034	3.68
$\beta=1, \alpha=32$	Semantic			Anti-steganalysis		
	Mauve ↑	BLEU ↑	Δ LDA ↓	Score ↑	TS_CSW ↓	UP4LS ↓
LLaMA2-7B	76.97	77.85	0.022	65.22	54.05	71.97
LLaMA2-13B	88.38	72.31	0.019	61.66	54.73	71.12

According to Table 9, it is not possible to generate stegos with stronger concealment by using larger LLMs. This is because steganography needs to embed secrets. The stego quality is more relevant with the embedding way rather than using larger LLMs.

5. Conclusion

To enhance the concealment of stegos, this paper proposes a steganography scheme based on allocated roulette area, namely DAIRstega. It uses the idea of the non-uniform roulette wheel and takes the CP of tokens as an important basis for allocating the number of codings. Then, the allocation function and three constraints are designed to optimize the allocation process. Experiments show that the proposed embedding way and DAIRstega have achieved excellent performance in various concealment and anti-steganalysis. Last but not least, to show the proposed scheme concisely, we did not conduct the results of all allocation functions and LLMs.

6. Outlook

By reflecting on this field, we found gaps in it that can be studied by scholars. **1.** This paper preliminarily explores the performance of the proposed method in long stegos. However, the research on long stegos is still in the basic stage. So, given the limited studies on longer and more diverse steganography, the scheme can be designed to improve the length and diversity. **2.** There is currently a lack of high-quality steganography schemes for closed-source LLMs. So, knowledge Graphs or Chain of Thought can enhance the prompts, while encoding text features allows for convenient and completely concealed steganography. **3.** The scheme in this paper can also be used as a watermarking for LLMs, which can make up for the insufficiency of the watermarking for LLMs being difficult to completely extract. Therefore, a new watermarking for LLMs can be designed by referring to the scheme in this paper. **4.** At present, linguistic steganography can achieve strong concealment and anti-steganalysis capabilities, that is, it is relatively safe. However, the sender of the stego is dangerous. Therefore, virtual social robots can be developed that utilize secrets to generate multi-turn conversations, promising contactless, covert communication that ensures the sender's safety. **5.** Finally, we found that existing steganalysis methods have poor detection performance on DAIRstega's stegos. Therefore, it is essential to advance steganalysis techniques.

Acknowledgment

Thank the editor and reviewers of “*Applied Soft Computing*” for their review and constructive suggestions.

Declaration of competing interest

The authors declare no conflict of interest.

Funding

This work is supported by the National Natural Science Foundation of China (Grant U21B2020) and supported by BUPT Excellent Ph.D. Students Foundation (Grant CX2023120, CX20241055).

Author contributions

Yihao Wang: Conceptualization; Data curation; Investigation; Methodology; Software; Funding acquisition; original draft. Ruiqi Song: Data curation; Investigation; Resources; Software; Validation; original draft. Lingxiao Li: Data curation; Resources; Software; Funding acquisition; original draft; review & editing. *Yihao Wang and Lingxiao Li contributed equally to this work and should be considered co-first authors.* Ru Zhang: Funding acquisition; Investigation; Project administration; Supervision; review & editing. Jianyi Liu: Formal analysis; Supervision; Investigation; Validation; review & editing.

Ethical Statement

This research does not involve human participants, personal data, or animal subjects; therefore, it does not raise any direct ethical concerns. The data used in this study were sourced from publicly available datasets or generated by applicable laws and regulations.

References

- [1] C. E. Shannon, Communication theory of secrecy systems, *The Bell system technical journal* 28 (4) (1949) 656–715.
- [2] T. Yang, H. Wu, B. Yi, G. Feng, X. Zhang, Semantic-preserving linguistic steganography by pivot translation and semantic-aware bins coding, *IEEE Transactions on Dependable and Secure Computing* (2023) 139–152.
- [3] Z. Yang, S. Zhang, Y. Hu, Z. Hu, Y. Huang, Vae-stega: linguistic steganography based on variational auto-encoder, *IEEE Transactions on Information Forensics and Security* 16 (2021) 880–895.
- [4] T. Lu, G. Liu, R. Zhang, T. Ju, Neural linguistic steganography with controllable security, in: *Proceeding of the Joint Conference on Neural Networks*, 2023, pp. 1–8.
- [5] H. Wang, Z. Yang, J. Yang, Y. Gao, Y. Huang, Hi-stega: A hierarchical linguistic steganography framework combining retrieval and generation, in: *Proceeding of International Conference on Neural Information Processing (ICONIP)*, 2023, pp. 41–54.
- [6] L. Huo, Y. Xiao, Synonym substitution-based steganographic algorithm with vector distance of two-gram dependency collocations, in: *Proceeding of International Conference on Computer and Communications*, 2016, pp. 2776–2780.
- [7] M. Kim, O. Zaiane, R. Goebel, Natural language watermarking based on syntactic displacement and morphological division, in: *Proceeding of the Annual IEEE Computer Software and Applications Conference*, 2010.
- [8] Y. Li, J. Zhang, Z. Yang, R. Zhang, Topic-aware neural linguistic steganography based on knowledge graphs, *ACM Transactions on Data Science* (2021) 1–13.
- [9] Z. Yang, X. Guo, Z. Chen, Y. Huang, Y. Zhang, Rnn-stega: Linguistic steganography based on recurrent neural networks, *IEEE Transactions on Information Forensics and Security* 14 (5) (2019) 1280–1295.
- [10] C. Ding, Z. Fu, Q. Yu, F. Wang, X. Chen, Joint linguistic steganography with bert masked language model and graph attention network, *IEEE Transactions on Cognitive and Developmental Systems* (2023) [doi:10.1109/TCDS.2023.3296413](https://doi.org/10.1109/TCDS.2023.3296413).
- [11] X. Zhou, W. Peng, B. Yang, J. Wen, Y. Xue, P. Zhong, Linguistic steganography based on adaptive probability distribution, *IEEE Transactions on Dependable and Secure Computing* 19 (5) (2021) 2982–2997.
- [12] S. Zhang, Z. Yang, J. Yang, Y. Huang, Provably secure generative linguistic steganography, in: *Proceeding of the International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 2021, pp. 3046–3055.
- [13] G. Kaptchuk, T. M. Jois, M. Green, A. D. Rubin, Meteor: Cryptographically secure steganography for realistic distributions, in: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 1529–1548.
- [14] Z. Yang, Y. Huang, Y. Zhang, A fast and efficient text steganalysis method, *IEEE Signal Processing Letters* 26 (4) (2019) 627–631.
- [15] Y. Tang, Y. Wang, R. Zhang, J. Liu, Linguistic steganalysis via llms: Two modes for efficient detection of strongly concealed stego., *IEEE Signal Processing Letters* 32 (2024) 541–545.
- [16] T. Fang, M. Jaggi, K. Argyraki, Generating steganographic text with lstms, in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)- Student Research Workshop*, 2017, pp. 100–106.
- [17] J. Ding, K. Chen, Y. Wang, N. Zhao, W. Zhang, N. Yu, Discop: Provably secure steganography in practice based on distribution copies, in: *IEEE Symposium on Security and Privacy (SP)*, 2023, pp. 2238–2255.
- [18] OpenAI, Gpt-4 technical report, arXiv preprint (2023). [arXiv:arXiv:2303.08774](https://arxiv.org/abs/2303.08774).
- [19] A. Alsajri, H. A. Salman, A. Steiti, Generative models in natural language processing: A comparative study of chatgpt and gemini, *Babylonian Journal of Artificial Intelligence* 2024 (2024) 134–145.
- [20] bdulazeez Alsajri, M. Aljanabi, The impact of chatgpt on social media, *MEDAAD* 2024 (2024) 9–14.
- [21] Meta, Llama 2: Open foundation and fine-tuned chat models, arXiv preprint (2023). [arXiv:arXiv:2307.09288](https://arxiv.org/abs/2307.09288).
- [22] P. Liu, P. Cheng, F. Li, W. Du, H. Zhao, G. Liu, Plmmark: a secure and robust black-box watermarking framework for pre-trained language models., in: *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, 2023, pp. 14991–14999.
- [23] X. Wu, C. Li, R. Y. Aminabadi, Z. Yao, Y. He, Understanding int4 quantization for language models: latency speedup, composability, and failure cases, in: *Proceedings of International Conference on Machine Learning (ICML)*, 2023, pp. 37524–37539.
- [24] X. Cai, J. Huang, Y. Bian, K. Church, Isotropy in the contextual embedding space: Clusters and manifolds, in: *Proceedings of International Conference on Machine Learning (ICML)*, 2021.
- [25] K. Pillutla, S. Swayamdipta, R. Zellers, J. Thickstun, S. Welleck, Mauve: Measuring the gap between neural text and human text using divergence frontiers, in: *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS)*, 2021, pp. 4816–4828.
- [26] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.
- [27] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, Bertscore: Evaluating text generation with bert, in: *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS)*, 2020.
- [28] A. Alsajri, A. Steiti, Intrusion detection system based on machine learning algorithms: (svm and genetic algorithm), *Babylonian Journal of Artificial Intelligence* 2024 (2024) 15–29.
- [29] Y. Wang, R. Zhang, J. Liu, Rls-dts: Reinforcement-learning linguistic steganalysis in distribution-transformed scenario, *IEEE Signal Processing Letters* 30 (2023) 1232–1236.
- [30] J. Wen, X. Zhou, P. Zhong, Y. Xue, Convolutional neural network based text steganalysis, *IEEE Signal Processing Letters* 26 (3) (2019) 460–464.
- [31] Z. Yang, Y. Huang, Y. Zhang, Ts-csw: text steganalysis and hidden capacity estimation based on convolutional sliding windows, *Multimedia Tools and Applications* 79 (2020) 18293–18316.
- [32] Q. Xu, R. Zhang, J. Liu, Linguistic steganalysis by enhancing and integrating local and global features, *IEEE Signal Processing Letters* 30 (2023) 16–20.
- [33] Y. Wang, R. Song, L. Li, Y. Tang, R. Zhang, J. Liu, Up4ls: User profile constructed by multiple attributes for enhancing linguistic steganalysis, arXiv preprint (2023). [arXiv:arXiv:2311.01775](https://arxiv.org/abs/2311.01775).
- [34] Y. Niu, J. Wen, P. Zhong, Y. Xue, A hybrid r-bilstm-c neural network based text steganalysis., *IEEE Signal Processing Letters* 26 (12) (2019) 107–1911.

- [35] J. Zou, Z. Yang, S. Zhang, S. Rehman, Y. Huang, High-performance linguistic steganalysis, capacity estimation and steganographic positioning., in: Proceeding of the International Workshop on Digital Watermarking (IWDW), 2021, pp. 80–93.
- [36] Y. Xu, T. Zhao, P. Zhong, Small-scale linguistic steganalysis for multi-concealed scenarios, *IEEE Signal Processing Letters* 29 (2022) 130–134.
- [37] J. Yang, Z. Yang, S. Zhang, H. Tu, Y. Huang, Sesy: Linguistic steganalysis framework integrating semantic and syntactic features., *IEEE Signal Processing Letters* 29 (2021) 31–35.
- [38] M. Bai, J. Yang, K. Pang, H. Wang, Y. Huang, Towards next-generation steganalysis: Llms unleash the power of detecting steganography, *arXiv preprint* (2024). [arXiv:arXiv:2405.09090](https://arxiv.org/abs/2405.09090).
- [39] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, T. Goldstein, A watermark for large language models, in: *Proceedings of International Conference on Machine Learning (ICML)*, 2023.
- [40] R. Zhang, S. S. Hussain, P. Neekhara, F. Koushanfar, Remark-llm: A robust and efficient watermarking framework for generative large language models, in: *Proceedings of Usenix security*, 2024.

7. Appendix

7.1. Supplement the effect of the proposed embedding way

This section gives the detailed values of Table 1, including the performance comparison of different embedding ways under different parameters and corresponding Bpw. The specific comparison results on Text quality, Statistical analysis, and Discourse matching are shown in Table 10, and the specific comparison results on anti-steganalysis are shown in Table 11.

Table 10: Text quality, statistical analysis, and discourse-matching comparison of stegos generated by DAIRstega and baselines. The embedding rates of each way include 1 Bpw (bit per word) and 2 Bpw. “Avg.” represents the overall performance. “Cap” represents the candidate pool. The meanings of “bit” are the same in Table 2. **Bold** is the best overall result. “*” is the suboptimal overall result. “↑” is that the higher the value, the better the result. “↓” means the lower the value, the better the result. For a detailed description of the embedding ways, please see section 4.1.

Embedding	Param / Bpw	Text quality		Statistical analysis					Discourse-matching			
		PPL ↓	ΔPcs ↓	CS ↑	JSD ↓	ED ↓	MD ↓	ADP ↓	ΔLDA ↓	Mauve ↑	BLEU ↑	Score ↑
LLMs+HC	bit = 1 / 1.00	13.331	5.639	96.54	45.79	0.263	8.255	3.46	0.014	70.60 _{±27.36}	54.26 _{±4.25}	58.44 _{±4.07}
	bit = 2 / 2.52	24.334	16.641	90.57	47.35	0.350	9.354	9.43	0.009	64.88 _{±30.03}	55.00 _{±5.77}	56.19 _{±4.01}
	Avg.	18.832	11.140	93.55	46.57	0.307	8.805	6.45	0.011	67.74	54.63	57.32
LLMs+ADG	$\tau = 0.3 / 1.34$	12.750	5.058	97.20	45.59	0.240	8.316	2.80	0.010	77.86 _{±23.65}	71.79 _{±3.03}	59.18 _{±5.84}
	$\tau = 0.65 / 2.99$	16.175	8.483	95.95	44.74	0.331	8.478	4.05	0.018	75.60 _{±27.02}	62.60 _{±4.68}	58.60 _{±5.69}
	Avg.	<u>14.463*</u>	<u>6.770*</u>	96.58	<u>45.17*</u>	0.285	<u>8.397*</u>	3.42	0.014	<u>76.73*</u>	<u>67.20*</u>	<u>58.35*</u>
LLMs+AC	Cap = 8 / 1.25	73.454	65.762	82.16	51.64	0.746	15.415	17.84	0.014	35.64 _{±22.65}	50.92 _{±29.30}	45.10 _{±3.07}
	Cap = 32 / 1.32	86.238	78.545	80.38	51.21	0.829	14.194	19.62	0.012	37.18 _{±24.59}	48.01 _{±27.45}	47.51 _{±2.76}
	Avg.	79.846	72.154	81.27	51.43	0.788	14.805	18.73	<u>0.013*</u>	36.41	49.47	46.31
LLMs+FLC	bit = 1 / 1.34	12.883	5.190	97.04	46.56	0.241	8.280	2.96	0.031	79.48 _{±26.53}	54.88 _{±6.33}	58.44 _{±4.64}
	bit = 2 / 2.67	30.481	22.788	93.86	46.30	0.328	8.939	6.14	0.011	61.10 _{±29.30}	58.21 _{±9.88}	55.20 _{±3.39}
	Avg.	21.682	13.989	95.45	46.43	<u>0.284*</u>	8.610	4.55	0.021	70.29	56.55	56.82
LLMs+Ours	$\beta=1 \alpha=8 / 1.10$	7.686	0.006	97.20	38.86	0.237	6.458	2.80	0.017	79.91 _{±26.75}	77.88 _{±30.40}	65.76 _{±4.13}
	$\beta=1 \alpha=32 / 1.11$	8.184	0.492	97.65	38.48	0.217	6.293	2.35	0.022	76.97 _{±26.47}	77.85 _{±30.42}	65.22 _{±4.61}
	$\beta=1 \alpha=48 / 1.13$	8.219	0.527	97.45	38.46	0.226	6.273	2.55	0.020	78.70 _{±25.51}	76.53 _{±31.71}	65.25 _{±4.78}
	Avg. (1Bpw)	8.030	0.342	97.43	38.60	0.227	6.341	2.57	0.020	78.53	77.42	65.41
	$\beta=0.5 \alpha=8 / 1.89$	13.775	6.082	97.39	37.80	0.229	7.204	2.61	0.018	80.83 _{±25.15}	73.19 _{±26.93}	62.49 _{±4.48}
	$\beta=0.5 \alpha=32 / 2.56$	20.876	13.184	92.43	39.69	0.389	8.976	7.57	0.029	75.58 _{±20.11}	66.99 _{±27.08}	60.89 _{±3.85}
	$\beta=0.5 \alpha=48 / 2.58$	22.069	14.377	92.44	38.31	0.389	8.592	7.56	0.006	72.40 _{±26.33}	79.64 _{±12.47}	60.29 _{±4.00}
	Avg. (2Bpw)	18.907	11.214	94.09	38.60	0.336	8.258	5.91	0.018	76.27	73.27	61.22
	Avg.	13.468	5.778	<u>95.76*</u>	38.60	0.281	7.299	<u>4.24*</u>	0.019	77.40	75.35	63.32

1. HC (Huffman coding) is used in VAE-Stega [3], LSCS [4], RNNstega [9], GAN-APD [11], JLS [10], and Discop [17].

2. ADG (Adaptive dynamic grouping) is used in Hi-Stega [5] and ADG [12].

3. AC (Arithmetic coding) is used in VAE-Stega [3], Hi-Stega [5], and Meteor [13].

4. FLC (Fixed length coding) is used in RNNstega [9] and JLS [10].

Table 11: Anti-steganalysis comparison of stegos generated by DAIRstega and baselines. The embedding rates of each way include 1Bpw (bit per word) and 2Bpw. “Avg.” represents the overall performance. **Bold** is the best overall result. “*” is the suboptimal overall result. “Cap” represents the candidate pool. “↑” is that the higher the value, the better the result. “↓” means the lower the value, the better the result. For a detailed description of the embedding ways, please see section 4.1.

Embedding	Param / Bpw	LS_CNN [30]		TS_CSW [31]		EILG [32]		UP4LS [33]	
		Acc ↓	F1 ↓	Acc ↓	F1 ↓	Acc ↓	F1 ↓	Acc ↓	F1 ↓
LLMs+HC	bit = 1 / 1.00	77.62 \pm 1.38	78.41 \pm 1.65	78.09 \pm 1.33	78.25 \pm 1.21	78.58 \pm 1.69	78.29 \pm 2.24	86.73 \pm 1.32	86.13 \pm 0.50
	bit = 2 / 2.52	77.54 \pm 1.50	77.57 \pm 2.05	76.03 \pm 1.74	76.09 \pm 1.33	78.29 \pm 0.72	78.95 \pm 1.47	88.19 \pm 1.00	88.43 \pm 2.61
	Avg.	77.58	77.99	77.06	77.17	78.44	78.62	87.46	87.28
LLMs+ADG	τ = 0.3 / 1.34	70.10 \pm 0.66	70.17 \pm 0.89	69.45 \pm 2.05	69.34 \pm 2.53	70.99 \pm 3.15	70.49 \pm 1.11	84.00 \pm 0.83	84.23 \pm 1.07
	τ = 0.65 / 2.99	63.06 \pm 2.49	63.69 \pm 3.61	62.22 \pm 1.46	61.86 \pm 1.18	66.14 \pm 1.38	66.15 \pm 1.38	80.42 \pm 0.59	80.57 \pm 0.58
	Avg.	66.58*	66.93*	65.84*	65.60*	68.57*	68.32*	82.21*	82.40*
LLMs+AC	Cap = 8 / 1.25	90.15 \pm 1.17	89.98 \pm 1.21	89.69 \pm 0.93	89.37 \pm 0.88	86.92 \pm 1.12	86.52 \pm 0.96	93.62 \pm 0.89	93.37 \pm 0.40
	Cap = 32 / 1.32	88.45 \pm 1.38	88.44 \pm 1.25	90.80 \pm 1.97	90.52 \pm 2.18	86.44 \pm 2.47	85.86 \pm 3.10	94.03 \pm 0.64	94.16 \pm 0.95
	Avg.	89.30	89.21	90.25	89.95	86.68	86.19	93.83	93.77
LLMs+FLC	bit = 1 / 1.34	78.53 \pm 1.20	80.05 \pm 1.39	82.72 \pm 1.27	83.32 \pm 1.10	76.67 \pm 0.85	76.15 \pm 1.96	89.03 \pm 0.25	89.20 \pm 0.80
	bit = 2 / 2.67	76.49 \pm 1.15	76.54 \pm 1.13	77.08 \pm 2.06	77.26 \pm 1.75	76.95 \pm 1.69	76.46 \pm 1.74	90.62 \pm 0.89	90.37 \pm 0.40
	Avg.	77.51	78.30	79.90	80.29	76.81	76.31	89.83	89.79
Ours	$\beta=1$ $\alpha=8$ / 1.10	59.50 \pm 2.77	60.15 \pm 10.59	56.82 \pm 4.09	64.21 \pm 10.03	67.25 \pm 3.23	68.04 \pm 1.53	75.40 \pm 0.37	76.26 \pm 1.28
	$\beta=1$ $\alpha=32$ / 1.11	59.35 \pm 1.25	59.99 \pm 5.51	54.05 \pm 4.72	61.68 \pm 11.17	62.28 \pm 0.50	62.66 \pm 2.31	71.97 \pm 0.64	72.75 \pm 1.21
	$\beta=1$ $\alpha=48$ / 1.13	58.00 \pm 2.37	56.08 \pm 13.79	54.98 \pm 3.14	59.17 \pm 11.37	65.51 \pm 2.23	66.78 \pm 3.91	71.40 \pm 0.65	70.89 \pm 2.05
	Avg. (1Bpw)	58.95	58.74	55.28	61.69	65.01	65.83	72.92	73.30
	$\beta=0.5$ $\alpha=8$ / 1.89	62.88 \pm 1.53	62.21 \pm 1.83	63.13 \pm 1.44	59.77 \pm 5.36	65.67 \pm 1.56	65.84 \pm 2.20	69.04 \pm 0.36	68.23 \pm 2.50
	$\beta=0.5$ $\alpha=32$ / 2.56	72.06 \pm 1.17	71.78 \pm 1.21	68.73 \pm 1.29	68.19 \pm 3.44	71.71 \pm 1.95	70.34 \pm 2.41	76.41 \pm 2.56	75.90 \pm 2.73
	$\beta=0.5$ $\alpha=48$ / 2.58	70.17 \pm 1.81	70.40 \pm 3.11	69.48 \pm 0.68	67.80 \pm 1.98	70.64 \pm 1.61	71.00 \pm 2.23	79.46 \pm 0.56	78.28 \pm 1.13
	Avg. (2Bpw)	68.37	68.13	67.11	65.25	69.34	69.06	74.97	74.14
	Avg.	63.66	63.44	61.20	63.47	67.18	67.44	73.95	73.72

1. HC (Huffman coding) is used in VAE-Stega [3], LSCS [4], RNNstega [9], GAN-APD [11], JLS [10], and Discop [17].
2. ADG (Adaptive dynamic grouping) is used in Hi-Stega [5] and ADG [12].
3. AC (Arithmetic coding) is used in VAE-Stega [3], Hi-Stega [5], and Meteor [13].
4. FLC (Fixed length coding) is used in RNNstega [9] and JLS [10].

7.2. Complete data for Table 5 and Table 6

In this section, we give the complete data of Table 5 and Table 6, as shown in Table 12 and Table 13. Table 12 contains the standard deviations of many datasets with different discourse, and Table 13 contains the standard deviations of multiple runs.

Table 12: Complete data for Table 5. The meanings of “bin” and “bit” are the same in Table 3. “a_{±b}” represents “Avg._{±Std}”. **Bold** is the best result. “*” is the suboptimal result. The “Std”s of DAIRstega Mauve and BLEU values are bigger. This is not due to the instability of 5-time runs, but is obtained from the effect on dozens of different discourse datasets, and represents the overall performance.

Schemes	Param / Bpw	Mauve ↑	BLEU ↑	Score ↑
Fang [16]	bin = 1 / 1.00	2.15 _{±0.87}	0.90 _{±1.58}	42.58 _{±1.30}
	bin = 3 / 3.00	1.96 _{±0.68}	0.90 _{±1.59}	41.65 _{±1.33}
RNNstega [9]	bit = 1 / 1.00	12.73 _{±10.56}	1.87 _{±3.28}	52.57 _{±2.20}
	bit = 3 / 2.63	11.97 _{±10.56}	1.83 _{±3.21}	52.36 _{±1.80}
ADG [12]	$\tau = 0.5 / 4.38$	18.54 _{±13.89}	2.72 _{±4.79}	47.55 _{±2.25}
LSCS [4]	- / 1.12	2.40 _{±1.78}	31.95 _{±12.89}	59.22 _{±3.14}
PLMmark [22]	- / -	7.22 _{±4.88}	0.15 _{±0.27}	43.50 _{±0.79}
DAIRstega	$\alpha = 8 / 1.10$	79.91 _{±26.75} *	76.37 _{±31.85}	65.76 _{±4.13}
	$\beta = 1$ $\alpha = 32 / 1.11$	76.97 _{±26.47}	77.85 _{±30.42} *	65.22 _{±4.61}
	$\alpha = 48 / 1.13$	78.70 _{±25.51}	76.53 _{±31.71}	65.25 _{±4.78} *
	$\alpha = 8 / 1.89$	80.83 _{±25.12}	73.19 _{±26.93}	62.49 _{±4.48}
	$\beta = 0.5$ $\alpha = 32 / 2.56$	75.58 _{±20.11}	66.99 _{±27.08}	60.89 _{±3.85}
	$\alpha = 48 / 2.58$	72.40 _{±26.33}	79.64 _{±12.47}	60.29 _{±4.00}

Table 13: Complete data for Table 6. The meanings of “bin” and “bit” are the same in Table 3. “a_{±b}” represents “Avg._{±Std}”. “Avg” and “Std” in this table are obtained by running 5 times. **Bold** is the best result. “*” is the suboptimal result.

Schemes	Param / Bpw	LS_CNN [30]		TS_CSW [31]		EILG [32]		UP4LS [33]	
		Acc ↓	F1 ↓	Acc ↓	F1 ↓	Acc ↓	F1 ↓	Acc ↓	F1 ↓
Fang [16]	bin = 1 / 1.00	99.70 _{±0.29}	99.68 _{±0.31}	88.34 _{±21.22}	91.01 _{±15.69}	99.50 _{±0.25}	99.51 _{±0.25}	99.75 _{±0.06}	99.60 _{±0.09}
	bin = 3 / 3.00	99.75 _{±0.16}	99.76 _{±0.15}	89.03 _{±20.58}	92.39 _{±13.82}	99.59 _{±0.17}	99.52 _{±0.66}	99.88 _{±0.10}	99.80 _{±0.15}
RNNstega [9]	bit = 1 / 1.00	85.81 _{±1.93}	86.03 _{±1.40}	81.39 _{±1.70}	82.59 _{±1.62}	85.86 _{±0.50}	86.29 _{±0.89}	97.42 _{±0.44}	95.97 _{±0.67}
	bit = 3 / 2.63	82.93 _{±2.18}	84.48 _{±2.20}	80.35 _{±2.38}	77.94 _{±3.04}	82.71 _{±0.17}	83.10 _{±1.02}	96.78 _{±0.82}	94.74 _{±1.34}
ADG [12]	$\tau = 0.5 / 4.38$	85.86 _{±1.73}	86.35 _{±1.82}	82.23 _{±2.40}	82.08 _{±2.96}	84.86 _{±1.74}	84.10 _{±1.80}	98.34 _{±0.39}	97.41 _{±0.58}
LSCS [4]	- / 1.12	78.86 _{±3.07}	78.98 _{±2.83}	81.19 _{±6.62}	80.11 _{±8.26}	84.15 _{±0.41}	85.11 _{±0.79}	96.22 _{±2.08}	95.48 _{±3.11}
PLMmark [22]	- / -	98.64 _{±0.92}	98.60 _{±0.95}	98.81 _{±1.83}	98.75 _{±1.93}	99.92 _{±0.17}	99.92 _{±0.11}	99.85 _{±0.30}	99.85 _{±0.30}
DAIRstega	$\alpha = 8 / 1.10$	59.50 _{±2.77}	60.15 _{±10.59}	56.82 _{±4.09}	64.21 _{±10.03}	67.25 _{±3.23}	68.04 _{±1.53}	75.40 _{±0.37}	76.26 _{±1.28}
	$\beta = 1$ $\alpha = 32 / 1.11$	59.35 _{±1.25} *	59.99 _{±5.51} *	54.05 _{±4.72}	61.68 _{±11.17}	62.28 _{±0.50}	62.66 _{±2.31}	71.97 _{±0.64}	72.75 _{±1.21}
	$\alpha = 48 / 1.13$	58.00 _{±2.37}	56.08 _{±13.79}	54.98 _{±3.14} *	59.17 _{±11.37}	65.51 _{±2.23} *	66.78 _{±3.91}	71.40 _{±0.65} *	70.89 _{±2.05} *
	$\alpha = 8 / 1.89$	62.88 _{±1.53}	62.21 _{±1.83}	63.13 _{±1.44}	59.77 _{±5.36} *	65.67 _{±1.56}	65.84 _{±2.20} *	69.04 _{±0.36}	68.23 _{±2.50}
	$\beta = 0.5$ $\alpha = 32 / 2.56$	72.06 _{±1.17}	71.78 _{±1.21}	68.73 _{±1.29}	68.19 _{±3.44}	71.71 _{±1.95}	70.34 _{±2.41}	76.41 _{±2.56}	75.90 _{±2.73}
	$\alpha = 48 / 2.58$	70.17 _{±1.81}	70.40 _{±3.11}	69.48 _{±0.68}	67.80 _{±1.98}	70.64 _{±1.61}	71.00 _{±2.23}	79.46 _{±0.56}	78.28 _{±1.13}

7.3. Supplement data for anti-steganalysis


We also perform other 7 steganalysis methods (including non-BERT-based, BERT-based, and LLM-based steganalysis) to detect the stegos we generated, as shown in Table 14, where, [15] is the current method that has achieved SOTA (State-of-the-art) steganalysis performances.

Table 14: The supplemental anti-steganalysis capability of the DAIRstega from other 7 steganalysis works. **Bold** represents the best specific and Avg. result. “*” represents the suboptimal specific result.

Supplemental anti-steganalysis			VAE-Stega [3]							DAIRstega						
			AC			HC			Avg.	$\alpha = 8$		$\alpha = 16$		$\alpha = 32$		Avg.
			Movie	Twitter	News	Movie	Twitter	News		$\beta = 1$	$\beta = 0.5$	$\beta = 1$	$\beta = 0.5$	$\beta = 1$	$\beta = 0.5$	
non-BERT- based	[14]	Acc ↓	57.63	53.75	52.55	61.63	57.75	74.38	59.61	54.38	51.37	<u>51.60*</u>	56.97	52.35	59.50	54.38
		F1 ↓	50.65	34.63	46.31*	60.89	66.65	73.89	55.50	47.09	47.55	49.47	52.35	49.20	57.70	50.56
	[34]	Acc ↓	64.50	58.75	62.50	87.63	80.75	94.38	74.75	64.75	<u>61.05*</u>	61.88	67.42	63.83	68.73	64.61
		F1 ↓	66.91	56.46	64.95	88.03	82.14	94.33	75.47	63.85	<u>61.05*</u>	67.03	66.16	63.82	68.83	65.12
BERT- based	[35]	Acc ↓	87.38	74.03	92.25	92.75	86.81	97.38	88.43	70.63	<u>65.43*</u>	65.06	65.87	65.56	73.25	67.63
		F1 ↓	87.39	74.21	91.99	93.21	86.85	97.35	88.50	70.44	61.48	65.28	65.15*	67.53	74.27	67.36
	[36]	Acc ↓	90.75	78.75	95.25	95.25	88.38	98.13	91.08	70.38	<u>65.50*</u>	64.38	71.12	<u>65.50*</u>	74.25	68.52
		F1 ↓	90.74	78.75	95.25	95.25	88.36	98.12	91.08	70.34	65.50	64.27	70.99	<u>65.31*</u>	74.25	68.44
	[37]	Acc ↓	92.50	75.38	95.88	94.75	88.32	97.62	90.74	70.12	64.25	64.25	73.75	<u>66.50*</u>	73.88	68.79
		F1 ↓	92.19	73.20	95.62	94.53	88.26	97.60	90.23	72.31	66.19	<u>66.29*</u>	71.47	70.02	69.04	69.22
LLM- based	[38]	Acc ↓	89.63	78.13	93.13	88.63	76.38	86.75	85.44	63.38	<u>59.63*</u>	61.44	63.25	59.38	62.13	61.53
		F1 ↓	89.59	78.06	93.12	88.62	76.13	86.75	85.38	63.24	60.15	<u>60.59*</u>	63.13	59.15	61.86	61.35
	[15]	Acc ↓	97.38	92.38	98.38	98.13	93.88	98.88	96.50	75.38	72.63	<u>71.75*</u>	81.75	69.38	83.25	75.69
		F1 ↓	96.87	92.97	98.57	97.10	92.97	98.97	96.25	75.27	72.41	<u>71.35*</u>	81.55	69.32	83.15	75.51

7.4. Example of cross-modal secret information hiding

Table 15: Examples of longer stegos generated by DAIRstega using cross-modal information. Bitstream is the binary corresponding to the secret.

Secret&Bitsream	Instruction	Stegos	PPL	Δ LDA
Secret:  Bitstream: 100010010101 000001001110 010001110000 110100001010 000110100000 101000000000 000000000000 000000001101 010010010100 100001000100 010100100000 000000000000 000000000110 110000000000 00...(31520 bit)	Discuss the causes of the Great Depression.	The Great Depression was a severe worldwide economic depression that took place from 1929 to 1939. It was the longest and most severe depression ever experienced by the industrialized world, surpassing in overall duration the Great Depression of 1873. The Depression began in the United States after the stock market crash of October 1929, and peaked between 1930 and 1933. By some measures, the Great Depression was the longest economic depression in the history of the industrialized world. The Great Depression had many causes, including overproduction, lack of consumer demand, high tariffs, banking crisis, stock market crash, deflation, wage reductions, and the 1929 collapse of Worldwide prices for agricultural commodities. The collapse of agricultural prices resulted from overproduction, due to improvements in seed stock, fertilizers, pesticides, transportation, and farm equipment, as well as increases in acreage and livestock numbers.	2.561	0.021
	Write a general overview of quantum computing.	Quantum computing is a new and revolutionary form of computing that is based on the principles of quantum mechanics. Unlike traditional computing, which uses binary bits to represent information, quantum computing uses quantum bits (qubits) which can exist in a superposition of 0 and 1 simultaneously. This means that quantum computers can process information exponentially faster than traditional computers, and they can also solve certain types of problems that are intractable for traditional computers. Quantum computing has the potential to revolutionize many fields, including machine learning, cryptography, drug discovery, and weather forecasting. For example, quantum computers can be used to train machine learning models more efficiently, which could lead to more accurate and efficient models. They can also be used to break current encryption methods, which could have serious implications for cybersecurity. In the field of drug discovery, quantum computers can be used to simulate molecular interactions, which could lead to the discovery of new and more effective drugs. Finally, in the field of weather forecasting, quantum computers can be used to simulate the complex interactions between different weather systems, which could lead to more accurate and precise forecasts.	1.915	0.009
...		