A Comprehensive Survey on Linguistic Steganography: Methods, Countermeasures, Evaluation, and Challenges

Anonymous ACL submission

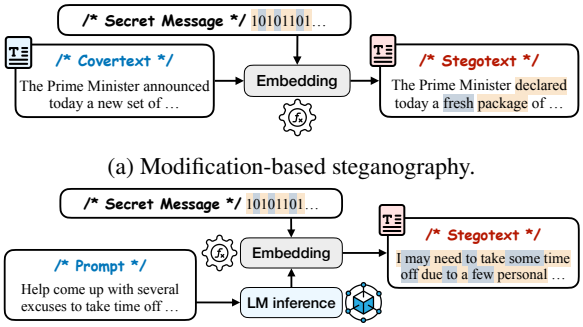
Abstract

Linguistic steganography, which hides secret messages in natural language text, plays an important role in information protection. Historically, its development was constrained by limited embedding capacity and text quality, but large language models (LLMs) have brought dramatic advances. Despite this progress, there is still a lack of a comprehensive survey that systematically covers linguistic steganography in the LLM era. In this work, we present a comprehensive survey with a focus on **LLM-era** developments, covering: (1) 140 modification-based and generative steganographic methods, (2) 58 countermeasures, i.e., linguistic steganalysis methods, (3) 23 evaluation metrics, and (4) 9 open challenges with future directions. For each of these four areas, we provide taxonomies, summaries, reviews, and adoption analyses. This survey aims to provide researchers with a systematic understanding of linguistic steganography in the LLM era and to guide future research toward more practical system designs.

1 Introduction

In recent decades, secure communication has been essential for protecting sensitive information and resisting surveillance (Merkle, 1978; Torrieri, 1992; Sutton, 2002; Wang et al., 2012). Cryptography and encryption systems are traditionally and widely used to safeguard messages by transforming them into unreadable forms that can only be deciphered with the proper key (Bhanot and Hans, 2015; Aumasson, 2024). However, data transmitted by these tools can be easily identifiable as encrypted communication, which can be detected and likely blocked in repressive environments (Raman et al., 2020; Kaptchuk et al., 2021), because such communication can be deemed suspicious.

To achieve both confidentiality and indistinguishability, researchers have explored alternative methods of covert communication that conceal not only the content of a message but also its existence. This pursuit has led to **steganography**, the practice of embedding hidden information within seemingly harmless carriers (Mishra and Bhanodiya, 2015). The roots of steganography can be traced back to antiquity, where civilizations such as the Greeks and Persians devised ingenious methods like carving messages on wooden tablets covered with wax or tattooing words on the scalp of messengers and waiting for the hair to regrow (Kahn, 1996). In the modern era, with the advancements of digital technology, steganography has been commonly based on digital carriers, including images (Subramanian et al., 2021), audio (Djebbar et al., 2012), video (Liu et al., 2019), and text (Majeed et al., 2021). Besides, steganographic techniques are also used in anonymous networks (Dingledine et al., 2004; Syverson et al., 1997).



(a) Modification-based steganography.

(b) Generative steganography based on language model(s).

Figure 1: An overview of how to achieve linguistic steganography based on two main frameworks.

Among these steganographic techniques, **linguistic steganography** (currently the dominant category in text steganography) has attracted increasing research attention for several reasons: (1) Textual content is ubiquitous on social networks (Aggarwal and Wang, 2011), making texts a natural and abundant carrier of hidden information. (2) Recent advances in large language models (LLMs) have

Survey	Year	Steganographic methods		Steganalysis methods		Evaluation metrics	
		Number	Categories (L1/L2/L3)	Number	Categories (L1/L2)	Number	Categories (L1/L2)
Majeed et al. (2021)	2021	50	3	–	–	–	–
Xiang et al. (2022b)	2022	37	4/11/–	–	–	11	4/–
Badawy et al. (2023)	2022	18	–	–	–	12	3/–
Wu et al. (2024a)	2023	28	2/4/8	29	2/7	8	–
Ours	**2026**	**140**	**2/6/17**	**58**	**2/8**	**23**	**3/6**

Table 1: Quantitative comparison of this survey with existing surveys, in which "–" indicates that relevant information on this dimension is not substantially or formally provided in the survey, and "L1/L2/L3" denotes the layer 1st/2nd/3rd for categorization.

Survey	Steganalysis taxonomy	Evaluation taxonomy	Robustness consideration	Analyses of targeted metrics	Analyses of metric adoption	Analyses of steganalysis adoption	Features of the LLM era
Majeed et al. (2021)	✗	✗	✓	✓	✗	✗	✗
Xiang et al. (2022b)	✗	✓	✗	✗	✗	✗	✗
Badawy et al. (2023)	✗	✓	✗	✗	✗	✓	✗
Wu et al. (2024a)	✓	✗	✗	✗	✗	✗	✗
Ours	✓	✓	✓	✓	✓	✓	✓

Table 2: Qualitative comparison of this survey with existing surveys.

significantly improved the fluency, coherence, and naturalness of generated steganographic texts (*stegotexts*) (Radford et al., 2019; Brown et al., 2020; Achiam et al., 2023), greatly surpassing traditional methods (Ziegler et al., 2019; Wu et al., 2024c). (3) Compared to steganography based on images, audio, or video (Cox et al., 1997), linguistic steganography is often less affected by compression-related distortion during transmission (Majeed et al., 2021). Figure 1 illustrates schematic diagrams of linguistic steganography, respectively, based on two primary frameworks: modification and generation.

In the era of large language models (LLMs), some influential open-weight models have been released, including the OPT series (Zhang et al., 2022), the LLaMA series (Touvron et al., 2023), and the Qwen series (Bai et al., 2023). These models have demonstrated strong performance across a wide range of downstream tasks, thereby enabling new possibilities for linguistic steganography. (Ding et al., 2024a; Zeng and Wang, 2025). However, a comprehensive study of linguistic steganography in the LLM era is still missing. In particular, previous surveys (Majeed et al., 2021; Xiang et al., 2022b; Badawy et al., 2023; Wu et al., 2024a) exhibit two main limitations:

- **Outdated.** They cover few techniques developed in the LLM era.
- **Incomplete.** Their analyses are coarse-grained and based on a narrow set of papers.

Motivated by these limitations, we present, to the best of our knowledge, **the first comprehensive survey of linguistic steganography in the context of LLMs**. This survey provides a broad coverage and systematic analyses of current linguistic steganographic **methods** based on modification and generation, **countermeasures** (steganalysis methods), **evaluation** metrics, and **challenges**. Table 1 presents quantitative comparisons, highlighting our contribution: a highly comprehensive, systematic LLM-era survey that covers about 3× more steganographic methods, about 2× more countermeasures (steganalysis), and about 2× more evaluation metrics than prior work. Table 2 presents qualitative comparisons, revealing our in-depth and fine-grained features. The main structure of this survey is shown in Figure 2. *To ensure both timeliness and comprehensiveness, this survey is motivated by the LLM era rather than being exclusively limited to the LLM era.* The survey is guided by the following research questions:

RQ1 (for methods): What is the complete landscape of linguistic steganography methods, and how can they be systematically categorized under modification-based and generative paradigms?

RQ2 (for methods): How have modern generative methods in the LLM era evolved with respect to key target metrics (*security, efficiency, and robustness*) and what are their features?

RQ3 (for countermeasures): What linguistic steganalysis countermeasures exist, how are they

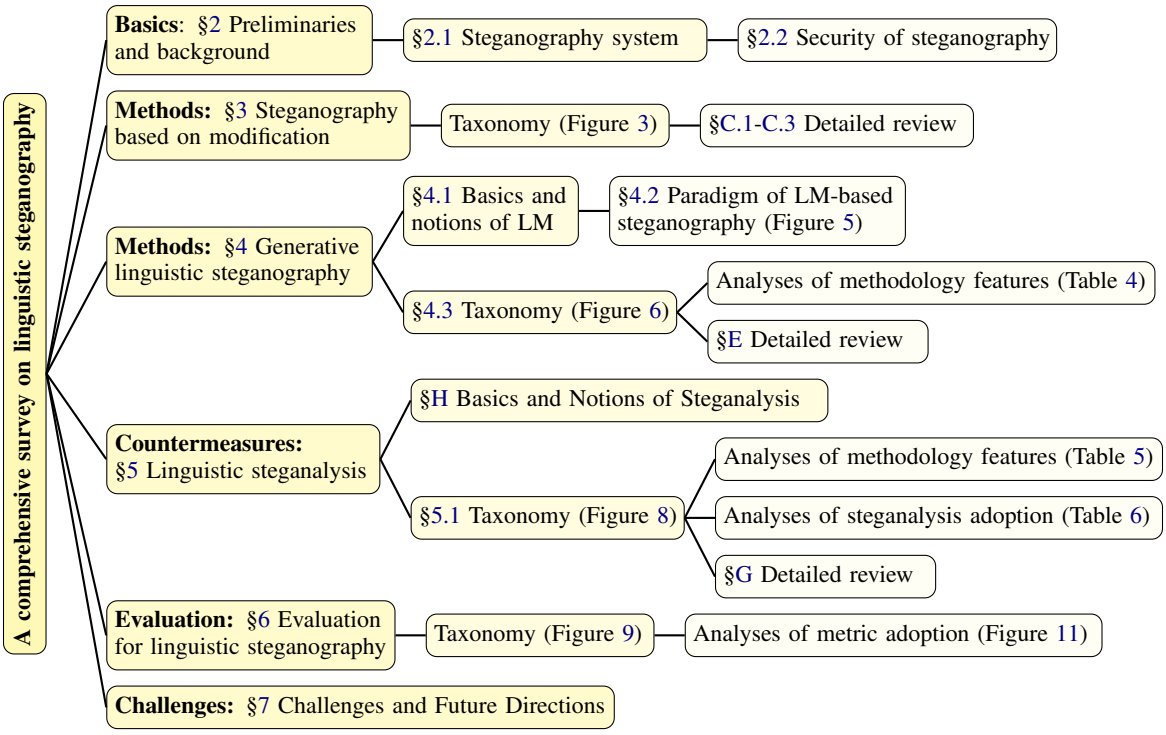


Figure 2: The main content flow and structure of this survey.

structured and modeled, what features do they possess, and how widely are they adopted?

RQ4 (for evaluation): What evaluation metrics are used in the field, how do they relate to the key metrics, and how widely are they adopted?

RQ5 (for challenges): What challenges remain in LLM-era linguistic steganography, and which directions are most promising for future research?

2 Preliminaries and Background

2.1 Steganography System

A classic scenario used to illustrate steganography is Simmons' *Prisoners' Problem* (Simmons, 1984). In this setting, Alice and Bob (the steganographers) are confined in prison and attempt to hatch an escape plan. Their only way of communication is monitored by the warden Eve (the steganalyzer), who monitors all messages. If Eve detects anything unusual or suspicious, she will block the exchange and impose punishment. Therefore, to evade detection, Alice and Bob must embed their secret messages into an *innocent-looking* cover object to obtain a stego object. Formally, a steganography system $\sum_{\mathcal{D}}$ with the channel distribution $\mathcal{D}$ is a triple of algorithms, $\sum_{\mathcal{D}} = (\text{KeyGen}_{\mathcal{D}}, \text{Embed}_{\mathcal{D}}, \text{Extract}_{\mathcal{D}})$, in which:

- $\text{KeyGen}_{\mathcal{D}}(1^{\lambda})$ takes arbitrary input with length λ and generates a shared key $K \in \{0,1\}^k$ with length k.
- $\text{Embed}_{\mathcal{D}}(K, \mathbf{m}, \mathcal{H})$ takes a shared key K, a secret message $\mathbf{m} \in \{0,1\}^*$, and a channel history $\mathcal{H}$ as the input. It returns the steganographic content, $\mathbf{s}$.
- $\text{Extract}_{\mathcal{D}}(K, \mathbf{s}, \mathcal{H})$ takes a shared key K, a steganographic content $\mathbf{s}$, and a channel history $\mathcal{H}$ as the input. It returns the message extracted from the steganographic content, $\mathbf{s}$.

To clarify the scope of steganography, we distinguish it from related concepts: cryptography and watermarking. The detailed comparison is shown in Appendix A.

2.2 Security of Steganography

To better understand linguistic steganography, it is important to first introduce a key property of steganographic systems: security. In the literature, two common definitions of steganographic security have been proposed. Cachin (1998) was the first to model steganographic security from an **information-theoretic** perspective. Hopper et al. (2002) and Katzenbeisser and Petitcolas (2002) proposed the **complexity-theoretic** definition of steganographic security. The detailed introduction of these two definitions is shown in Appendix B.

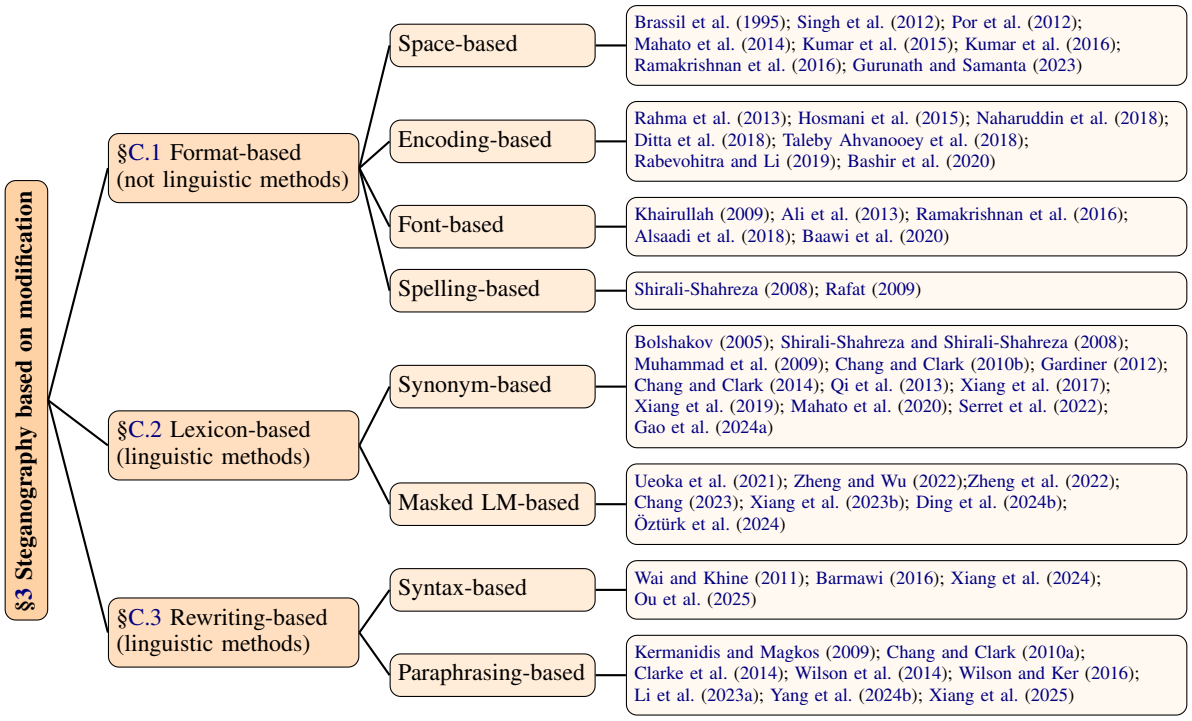


Figure 3: The main content flow and categorization of steganographic methods based on modification.

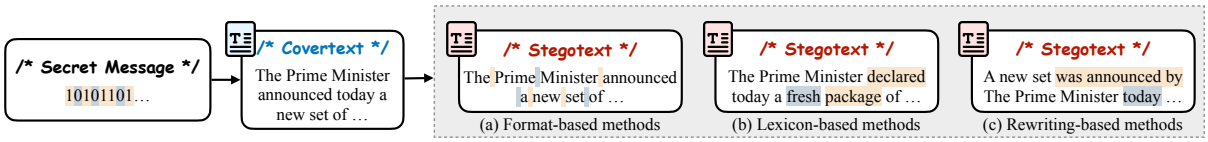


Figure 4: Modification-based steganographic methods which are categorized into three types, where an example is shown in each type.

3 Steganography Based on Modification

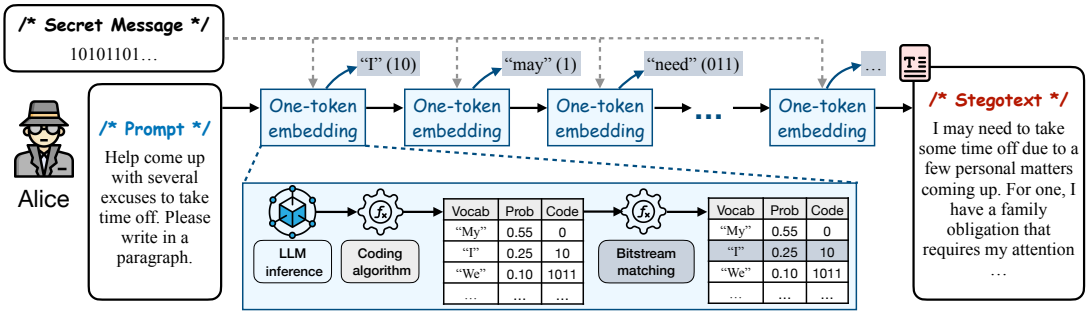
Text steganographic methods based on modifying existing texts had dominated the field before the emergence of generative LMs. These approaches embed secret messages through alterations to a covertext, while aiming to preserve its readability and naturalness. We begin with **non-linguistic** format-based methods (e.g., manipulating spaces, punctuation, or text layout) and then move to **linguistic** methods, including lexicon-based (e.g., replacing words with synonyms or semantically similar alternatives) and rewriting-based methods (e.g., paraphrasing or restructuring sentences). Figure 3 presents the content flow and taxonomy of modification-based methods in this section. Figure 4 shows an example in each category of these modification-based steganographic methods. The detailed introduction of methods based on modification is provided in Appendix C.

Although such methods are generally lightweight and easy to implement, they suffer from inherent limitations: the embedding capacity is low, the semantic information could be inevitably altered, and the modifications can be vulnerable to detection by statistical steganalysis. However, modification-based steganography has laid the foundation for subsequent advances and served as a baseline compared to which generative methods are evaluated (Ueoka et al., 2021).

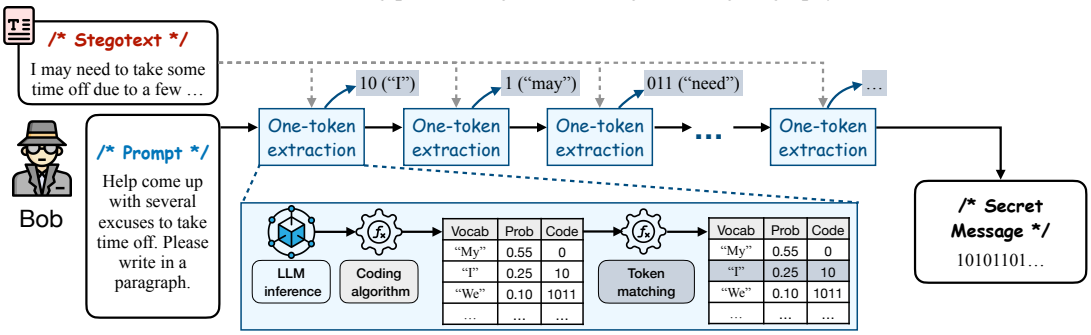
4 Generative Linguistic Steganography

As discussed in Section 3, existing modification-based methods generally face the challenge of low embedding capacity, as the redundancy in the covertext is restricted. Besides, in these methods, security is not proved or ensured according to the security definition discussed in Section 2.2.

These limitations have motivated a shift toward generative approaches, which offer greater flexibility and high imperceptibility. Some of them also provide information-theoretic security guarantees that focus mainly on achieving zero KL di-



(a) Embedding process of generative linguistic steganography.



(b) Extraction process of generative linguistic steganography.

Figure 5: Overall framework of generative linguistic steganography, in which the example is operated autoregressively based on the Huffman coding.

vergence caused by steganographic encoding in existing methods (Zhang et al., 2021b; Ding et al., 2023; Wang et al., 2025b). Generative linguistic steganography has attracted increasing research attention for several reasons: (1) high capacity, (2) advances of language models, (3) Imperceptibility and security. The detailed introduction of these reasons is provided in Appendix D.

4.1 Basics and Notions of LM

A language model (LM) has a vocabulary $\mathcal{V}$ consisting of $|\mathcal{V}|$ *tokens*. Consider a sequence of LM-generated T tokens $\{s^{(t)}\} \in \mathcal{V}^T$. Entries with negative indices, $[s^{(-N_p)}, \ldots, s^{(-1)}]$, represent a *prompt* of length N_p and $[s^{(0)}, \ldots, s^{(T-1)}]$ are tokens generated by an LM in response to the prompt.

Using an LM for the next-token prediction at position t, is a function defined as: $f_{\text{LM}}(\boldsymbol{s}^{(t-1)}; \theta) = \boldsymbol{l}^{(t)}$, in which $\boldsymbol{s}^{(t-1)}$ is the previous context $[s^{(-N_p)}, \ldots, s^{(t-1)}]$ at $t-1$, θ is the model parameters, and $\boldsymbol{l}^{(t)} = (l_1^{(t)}, \ldots, l_{|\mathcal{V}|}^{(t)}) \in \mathbb{R}^{|\mathcal{V}|}$ is the logit vector, corresponding to each token in $\mathcal{V}$. This discrete probability distribution can be calculated through the logit vector and the softmax function, i.e., $\boldsymbol{p}^{(t)} = (p_1^{(t)}, \ldots, p_{|\mathcal{V}|}^{(t)}) = \text{softmax}(\boldsymbol{l}^{(t)}) = (\frac{\exp(l_1^{(t)})}{\sum_{j=1}^{|\mathcal{V}|}\exp(l_j^{(t)})}, \ldots, \frac{\exp(l_{|\mathcal{V}|}^{(t)})}{\sum_{j=1}^{|\mathcal{V}|}\exp(l_j^{(t)})})$ over the vocabulary. The next token is then sampled from $\boldsymbol{p}^{(t)}$ (e.g., multinomial sampling).

4.2 Paradigm of LM-based Steganography

Alice (the sender) wants to communicate a secret message $\mathbf{m} \sim U(\{0,1\}^l)$ with Bob (the receiver) by embedding it in a natural-language text $\mathbf{s}$ (a stegotext). The uniform distribution is chosen for $\mathbf{m}$ without loss of generality: if $\mathbf{m}$ has additional structure it can be further compressed to a uniformly distributed random variable (Han, 2005). According to the notions and definitions of the steganographic system in Section 2.1, Alice and Bob have agreed on an embedding function $\text{Embed}_{\mathcal{D}}(K, \mathbf{m}, \mathcal{H})$ and an extracting function $\text{Extract}_{\mathcal{D}}(K, \mathbf{s}, \mathcal{H})$ that perform steganography. Specifically, in the scenario of LM-based generative linguistic steganography, $\mathcal{D}$ is commonly defined as the shared same language model $\mathcal{M}^o$, $\mathcal{H}$ is the token sequence of the prompt $[s^{(-N_p)}, \ldots, s^{(-1)}]$, $\mathbf{s}$ is the generated stegotext. These two functions are designed to be mutually consistent. In other words, $\text{Embed}_{\mathcal{D}}(K, \mathbf{m}, \mathcal{H}) = \mathbf{s}$ and $\text{Extract}_{\mathcal{D}}(K, \mathbf{s}, \mathcal{H}) = \mathbf{m}'$, in which the ideal result is $\mathbf{m} = \mathbf{m}'$.

Figures 5a and 5b respectively illustrate the embedding framework and the extraction framework of LM-based steganography, using the method

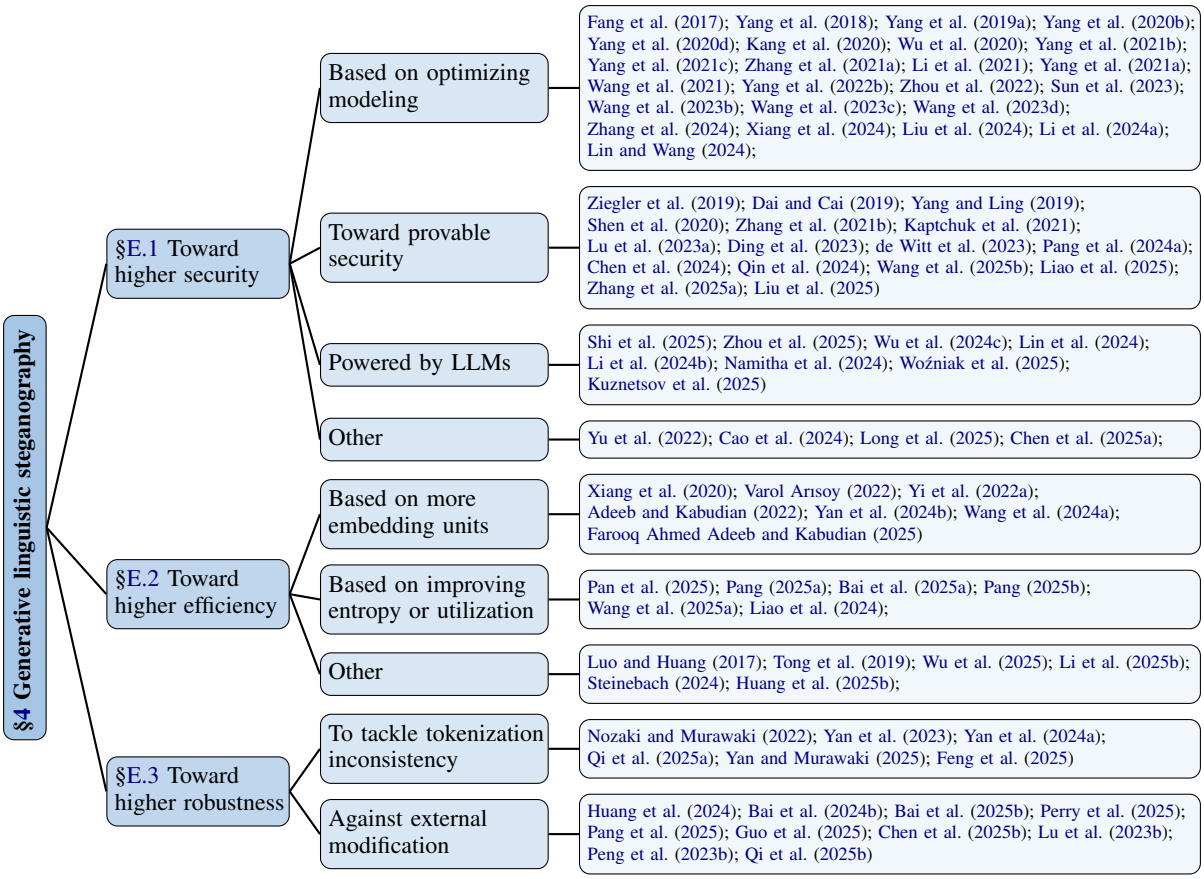


Figure 6: The main content flow and categorization of generative linguistic steganography.

based on the Huffman coding (Yang et al., 2019a). From the figures, we can notice that the cover carrier, which is necessary in traditional steganography, is not required in generative linguistic steganography. Instead, the prompt is shared in the steganographic protocol. Although generative linguistic steganographic methods generally follow the above frameworks, they can be classified according to the targeted performance metrics, methodology features, or the specific steganographic scenarios in which they are applied.

4.3 Taxonomy of Existing Methods

In this section, we categorize existing generative linguistic steganographic methods into three types: (1) methods toward higher security, (2) methods toward higher efficiency, and (3) methods toward higher robustness. **Security** can be assessed using various text-quality metrics (such as perplexity (Jelinek et al., 1977) and ROUGE (Lin, 2004)) or by measuring imperceptibility and indistinguishability that prevent humans or machine classifiers (steganalysis) from detecting stegotexts from normal texts. **Efficiency** refers to the embedding capacity of the stegotext as well as the embedding and extraction speeds. **Robustness** evaluates whether, and to which extent, the extracted message $\mathbf{m}'$ remains consistent with the embedded message $\mathbf{m}$, particularly under active attack scenarios, such as substitution, deletion, or insertion.

Following Fang et al. (2017), which marked a milestone for RNN-based generative linguistic steganography, only modern methods (after Fang et al. (2017)) are included for taxonomy. Figure 6 presents the main content flow and categorization of generative linguistic steganographic methods, organized according to the **primary targeted metrics** and further methodology features. The detailed introduction and the review of modern generative linguistic steganographic methods are provided in Appendix E. The review of early generative linguistic steganographic methods is provided in Appendix F.

5 Linguistic Steganalysis

Like many security technologies, linguistic steganography is dual-use in nature, which can be used not only to protect personal privacy but

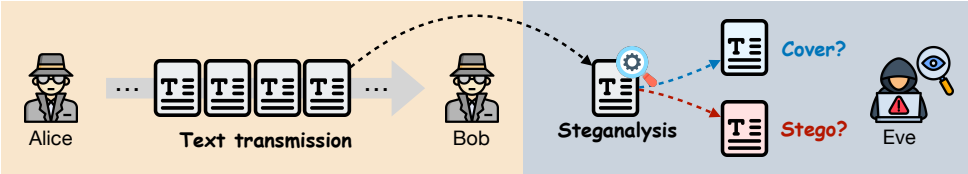


Figure 7: A sketch of how Eve operates steganalysis.

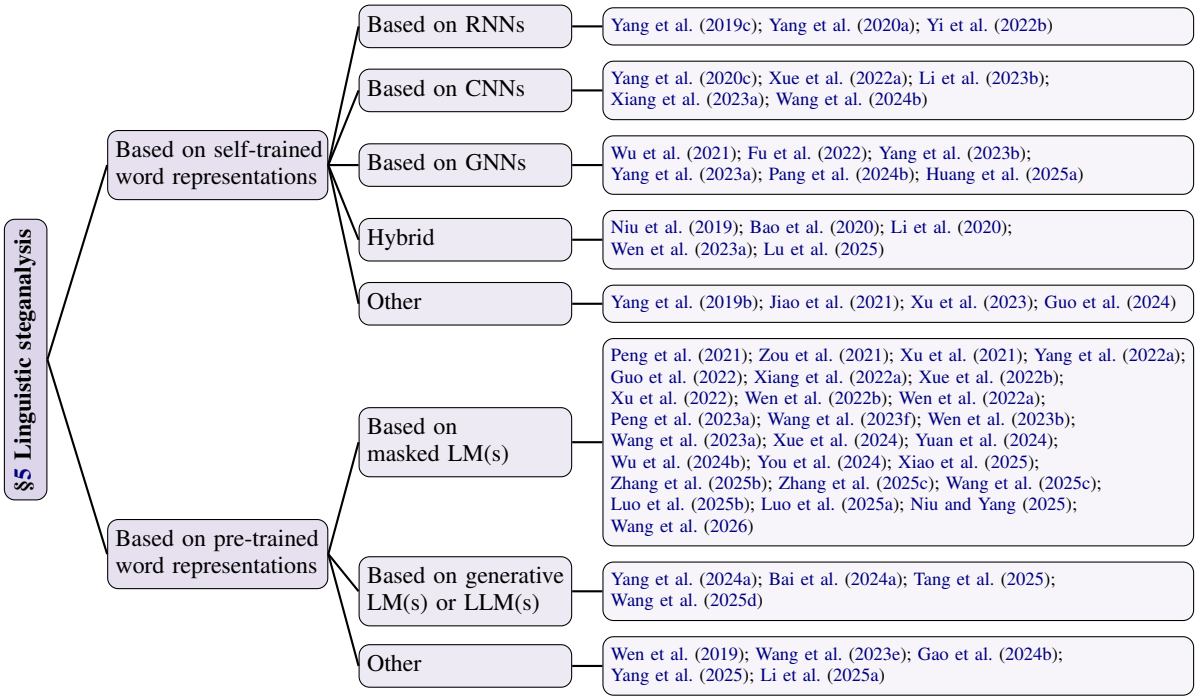


Figure 8: The main content flow and categorization of linguistic steganalysis methods.

also to enable malicious covert communication. Its countermeasures, **linguistic steganalysis** methods, are essential for detecting the presence of steganography, particularly in security-critical contexts (Fridrich and Goljan, 2002; Nissar and Mir, 2010). Figure 7 shows a sketch of steganalysis. The introduction of basics and notions of steganalysis is provided in Appendix H.

5.1 Taxonomy of Existing Methods

We categorize existing linguistic steganalysis methods along two axes. First, we distinguish whether a method relies on pre-trained word representations or models, such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), or BERT (Devlin et al., 2019). Second, we further group methods by the type of backbone model they adopt. Following these principles, Figure 8 presents the resulting taxonomy of linguistic steganalysis methods. To keep the taxonomy focused on recent developments, our taxonomy and review only includes methods based on neural networks that are proposed after current representative methods, **FCN** (Yang et al., 2019b) and **TS-RNN** (Yang et al., 2019c). Table 5 (in the Appendix) lists the features of these linguistic steganalysis methods, revealing the following trends:

(1) The **IMDB** (Maas et al., 2011), **Twitter** (Go et al., 2009), and **News** (Kaggle, 2017) datasets have become mainstream for evaluation.

(2) Chronologically, methods based on **pre-trained** models (especially BERT (Devlin et al., 2019)) gradually **dominate** the field.

(3) **LLM-based** linguistic steganalysis methods have emerged only recently and remain relatively **underexplored**.

In addition, Table 6 (in the Appendix) summarizes the steganalysis methods adopted to evaluate generative linguistic steganography. Among them, **FCN** (Yang et al., 2019b) **CNN** (Wen et al., 2019), **R-BiLSTM-C** (Niu et al., 2019), **TS-RNN** (Yang et al., 2019c), **TS-CSW** (Yang et al., 2020c) are the five most commonly adopted steganalysis methods. We also observe that most steganographic meth-

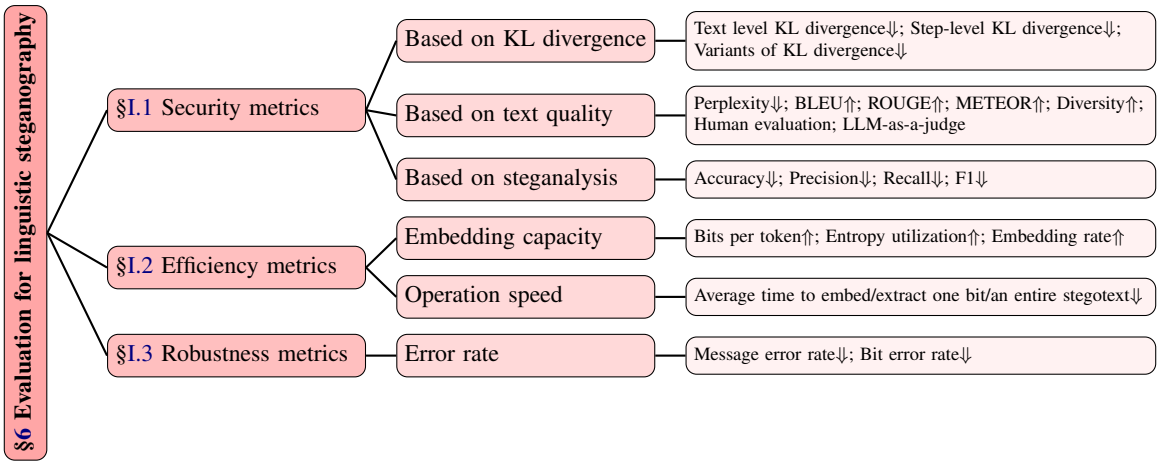


Figure 9: An overview of evaluation metrics. ⇑/⇓ indicate that higher/lower values correspond to better performance.

ods adopt **at least one** steganalysis model to assess imperceptibility. The detailed review of linguistic steganalysis methods are provided in Section G.

6 Evaluation for Linguistic Steganography

Following the taxonomy of generative linguistic steganography in Section 4.3, this section details the main metrics for evaluating linguistic steganography, which are categorized into security metrics, efficiency metrics, and robustness metrics.

Figure 9 presents an overview and their taxonomy of these evaluation metrics. Figure 11 (in the Appendix) summarizes the chronological trend on adoption rates of the major evaluation approaches in 87 papers of modern generative linguistic steganographic methods. Tables 7 and 8 (in the Appendix) present metrics chosen to evaluate these generative methods. From these tables, we draw the following key observations:

(1) PPL, metrics based on KL divergence, and BPT are used **most frequently**.

(2) Entropy utilization is gaining emphasis because model-agnostic methods compatible with diverse LLMs require **fair cross-model assessments** of embedding capacity.

(3) Metrics based on error rates are used increasingly with the growing focus on **robustness**.

The detailed introduction of these evaluation metrics is shown in Appendix I.

7 Challenges and Future Directions

Despite the rapid progress surveyed above, linguistic steganography is still far from being a mature, well-understood technology. One warning sign is the continued lack of ready-to-use software, despite the steady publication of relevant papers. To better understand the factors underlying these limitations, we outline several key challenges and promising future directions: (1) computational indeterminism, (2) limited embedding capacity, (3) tokenization inconsistency, (4) abrupt termination, (5) high dependence on white-box LLMs, (6) challenges of synchronizing prompts, (7) unrealistic security evaluations, (8) lack of unity in benchmarks, and (9) ethics and use restrictions of LLM licenses. The detailed introduction for these aspects is provided in Appendix J.

8 Conclusion

In this survey, we provide a comprehensive and systematic overview of linguistic steganography in the LLM era. We organize the content of existing work into three main axes: (1) modification-based and generative steganographic methods, (2) linguistic steganalysis techniques, and (3) evaluation metrics for security, efficiency, and robustness. They cover recent advances, especially including **provably secure** methods, **robustness-oriented** designs, and emerging **black-box** and **asymmetric** schemes. Building on the current features and trends of linguistic steganography in the LLM era, we highlight several open challenges and discuss ethical and licensing constraints surrounding LLM use relevant for real-world deployment. We hope that this survey can serve as both a reference and a roadmap, helping researchers design linguistic steganography systems that are computationally practical and responsibly used in the LLM era.

Limitations

This work is intended to be a comprehensive survey, rather than a survey restricted exclusively to LLM-era linguistic steganography. We recognize that this broader scope may, from the perspective of some readers, reduce the perceived timeliness or readability of the paper. However, we chose this scope for the following reasons:

(1) Existing pre-LLM surveys do not provide a truly comprehensive review of linguistic steganographic methods, as shown in Tables 1 and 2. This gap motivates us to systematically survey and review the full body of work, including both pre-LLM and LLM-era methods.

(2) Linguistic steganography, linguistic steganalysis, and evaluation methods have not yet been broadly reshaped by LLMs, as analyzed in Figures 6, 8, and 11. Restricting the survey to LLM-era methods exclusively would therefore result in an overly narrow scope.

(3) A substantial portion of linguistic steganography methods are *model-agnostic* and can be applied to both small and large language models. As a result, distinguishing methods solely by whether they are "LLM-era" is often of limited significance.

Ethical Considerations

Although linguistic steganography can serve legitimate purposes, such as embedding and tracing copyright information, it can also be misused to evade legal censorship. This dual-use nature underscores the urgent need for effective regulation. In this survey, we highlight the ethics and use restrictions (Section 7 and Appendix J) and advocate for research on both responsible defensive capabilities and legitimate privacy-preserving uses, thereby promoting responsible use of steganography.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Omer Farooq Ahmed Adeeb and Seyed Jahanshah Kabudian. 2022. Arabic text steganography based on deep learning methods. *IEEE Access*, 10:94403–94416.

Charu C. Aggarwal and Haixun Wang. 2011. *Text Mining in Social Networks*, pages 353–378. Springer US, Boston, MA.

Abdelmgeid Amin Ali and 1 others. 2013. New text steganography technique by using mixed-case font. *International Journal of Computer Applications*, 62(3).

Husam Ibrahiem Alsaadi, Maad Kamal Al-Anni, Rafah M. Almuttairi, Oguz Bayat, and Osman Nuri Ucan. 2018. Text steganography in font color of ms excel sheet. In *Proceedings of the First International Conference on Data Science, E-Learning and Information Systems*, DATA '18, New York, NY, USA. Association for Computing Machinery.

Berk Atil, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J. Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudrej, Ferhan Ture, Zhe Wu, Lixinyu Xu, and Breck Baldwin. 2025. Non-determinism of "deterministic" llm settings. *Preprint*, arXiv:2408.04667.

Jean-Philippe Aumasson. 2024. *Serious cryptography: a practical introduction to modern encryption*. No Starch Press, Inc.

Salwa Shakir Baawi, Dhamyaa A. Nasrawi, and Lina Talib abdulameer. 2020. Improvement of "text steganography based on unicode of characters in multilingual" by custom font with special properties. *IOP Conference Series: Materials Science and Engineering*, 870(1):012125.

Israa Lotfy Badawy, Khaled Nagaty, and Abeer Hamdy. 2023. A comprehensive review on deep learning-based generative linguistic steganography. In *Learning in the Age of Digital and Green Transition*, pages 651–660, Cham. Springer International Publishing.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Minhao Bai, Yongfeng Huang, Jinshuai Yang, Kaiyi Pang, and Songbin Li. 2024a. Exploration of the effectiveness and characteristics of chatgpt in steganalysis tasks. In *Proceedings of the 2023 4th Asia Service Sciences and Software Engineering Conference*, ASSE '23, page 163–170, New York, NY, USA. Association for Computing Machinery.

Minhao Bai, Kaiyi Pang, Guorui Liao, Jinshuai Yang, and Yongfeng Huang. 2025a. Shimmer: a provably secure steganography based on entropy collecting mechanism. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 5949–5965.

Minhao Bai, Jinshuai Yang, Kaiyi Pang, Yongfeng Huang, and Yue Gao. 2024b. Semantic steganography: A framework for robust and high-capacity information hiding using large language models. *Preprint*, arXiv:2412.11043.

Minhao Bai, Jinshuai Yang, Kaiyi Pang, Xin Xu, Zhen Yang, and Yongfeng Huang. 2025b. Provably robust

and secure steganography in asymmetric resource scenario. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 1438–1456.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

YongJian Bao, Hao Yang, ZhongLiang Yang, Sheng Liu, and YongFeng Huang. 2020. Text steganalysis with attentional l stm-cnn. In *2020 5th International Conference on Computer and Communication Systems (ICCCS)*, pages 138–142.

Ari Barmawi. 2016. Linguistic based steganography using lexical substitution and syntactical transformation. In *2016 6th International Conference on IT Convergence and Security (ICITCS)*, pages 1–6.

Hafsat Muhammad Bashir, Qianmu Li, and Jun Hou. 2020. A high capacity text steganography utilizing unicode zero-width characters. In *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, pages 668–675.

Richard Bergmair and Stefan Katzenbeisser. 2007. Content-aware steganography: About lazy prisoners and narrow-minded wardens. In *Information Hiding*, pages 109–123, Berlin, Heidelberg. Springer Berlin Heidelberg.

Rajdeep Bhanot and Rahul Hans. 2015. A review and comparative analysis of various encryption algorithms. *International Journal of Security and Its Applications*, 9(4):289–306.

Igor A. Bolshakov. 2005. A method of linguistic steganography based on collocationally-verified synonymy. In *Information Hiding*, pages 180–191, Berlin, Heidelberg. Springer Berlin Heidelberg.

J.T. Brassil, S. Low, N.F. Maxemchuk, and L. O'Gorman. 1995. Electronic marking and identification techniques to discourage document copying. *IEEE Journal on Selected Areas in Communications*, 13(8):1495–1504.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Christian Cachin. 1998. An information-theoretic model for steganography. In *Information Hiding*, pages 306–318, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chris Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 196–205, Honolulu, Hawaii. Association for Computational Linguistics.

Yi Cao, Zhili Zhou, Chinmay Chakraborty, Meimin Wang, Q. M. Jonathan Wu, Xingming Sun, and Keping Yu. 2024. Generative steganography based on long readable text generation. *IEEE Transactions on Computational Social Systems*, 11(4):4584–4594.

Ching-Chun Chang. 2023. Reversible linguistic steganography with bayesian masked language modeling. *IEEE Transactions on Computational Social Systems*, 10(2):714–723.

Ching-Yun Chang and Stephen Clark. 2010a. Linguistic steganography using automatically generated paraphrases. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 591–599, Los Angeles, California. Association for Computational Linguistics.

Ching-Yun Chang and Stephen Clark. 2010b. Practical linguistic steganography using contextual synonym substitution and vertex colour coding. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1194–1203, Cambridge, MA. Association for Computational Linguistics.

Ching-Yun Chang and Stephen Clark. 2012. The secret's in the word order: Text-to-text generation for linguistic steganography. In *Proceedings of COLING 2012*, pages 511–528, Mumbai, India. The COLING 2012 Organizing Committee.

Ching-Yun Chang and Stephen Clark. 2014. Practical linguistic steganography using contextual synonym substitution and a novel vertex coding method. *Computational Linguistics*, 40(2):403–448.

Mark Chapman and George Davida. 1997. Hiding the hidden: A software system for concealing ciphertext as innocuous text. In *Information and Communications Security*, pages 335–345, Berlin, Heidelberg. Springer Berlin Heidelberg.

Mark Chapman, George I. Davida, and Marc Rennhard. 2001. A practical and effective approach to large-scale automated linguistic steganography. In *Information Security*, pages 156–165, Berlin, Heidelberg. Springer Berlin Heidelberg.

Mark Chapman and George Davida2. 2002. Plausible deniability using automated linguistic stegonagraphy. In *Infrastructure Security*, pages 276–287, Berlin, Heidelberg. Springer Berlin Heidelberg.

Cheng Chen, Jinshuai Yang, Yue Gao, Huili Wang, and Yongfeng Huang. 2024. Minimizing distortion in steganography via adaptive language model tuning. In *Neural Information Processing*, pages 571–584, Singapore. Springer Nature Singapore.

Yingquan Chen, Qianmu Li, Aniruddha Bhattacharjya, Xiaocong Wu, Huifeng Li, Qing Chang, Le Zhu, and Yan Xiao. 2025a. Enhancement of the generation quality of generative linguistic steganographic texts by a character-based diffusion embedding algorithm (cdea). *Applied Sciences*, 15(17).

Yingquan Chen, Qianmu Li, Xiaocong Wu, and Zijian Ying. 2025b. Research on making two models based on the generative linguistic steganography for securing linguistic steganographic texts from active attacks. *Symmetry*, 17(9).

Charles A. Clarke, Eckhard Pfluegel, and Dimitris Tsaptsinos. 2014. Hide-as-you-type: An approach to natural language steganography through sentence modification. In *2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICESS)*, pages 945–952.

I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon. 1997. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687.

Jonathan Cui and David A. Araujo. 2024. Rethinking use-restricted open-source licenses for regulating abuse of generative models. *Big Data & Society*, 11(1).

Falcon Dai and Zheng Cai. 2019. Towards near-imperceptible steganographic text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4303–4308, Florence, Italy. Association for Computational Linguistics.

Weihui Dai, Yue Yu, Yonghui Dai, and Bin Deng. 2010. Text steganography system using markov chain source model and des algorithm. *J. Softw.*, 5(7):785–792.

Weihui Dai, Yue Yu, and Bin Deng. 2009. Bintext steganography based on markov state transferring probability. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, ICIS '09, page 1306–1311, New York, NY, USA. Association for Computing Machinery.

Christian Schroeder de Witt, Samuel Sokota, J Zico Kolter, Jakob Nicolaus Foerster, and Martin Strohmeier. 2023. Perfectly secure steganography using minimum entropy coupling. In *The Eleventh International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Changhao Ding, Zhangjie Fu, Zhongliang Yang, Qi Yu, Daqiu Li, and Yongfeng Huang. 2024a. Context-aware linguistic steganography model based on neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:868–878.

Changhao Ding, Zhangjie Fu, Qi Yu, Fan Wang, and Xianyi Chen. 2024b. Joint linguistic steganography with bert masked language model and graph attention network. *IEEE Transactions on Cognitive and Developmental Systems*, 16(2):772–781.

Jinyang Ding, Kejiang Chen, Yaofei Wang, Na Zhao, Weiming Zhang, and Nenghai Yu. 2023. Discop: Provably secure steganography in practice based on "distribution copies". In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2238–2255.

Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The Second-Generation onion router. In *13th USENIX Security Symposium (USENIX Security 04)*, San Diego, CA. USENIX Association.

Allah Ditta, Cai Yongquan, Muhammad Azeem, Khurram Gulzar Rana, Haiyang Yu, and Muhammad Qasim Memon. 2018. Information hiding: Arabic text steganography by using unicode characters to hide secret data. *International Journal of Electronic Security and Digital Forensics*, 10(1):61–78.

Fatiha Djebbar, Beghdad Ayad, Karim Abed Meraim, and Habib Hamam. 2012. Comparative study of digital audio steganography techniques. *EURASIP Journal on Audio, Speech, and Music Processing*, 2012(1):25.

Tina Fang, Martin Jaggi, and Katerina Argyraki. 2017. Generating steganographic text with LSTMs. In *Proceedings of ACL 2017, Student Research Workshop*, pages 100–106, Vancouver, Canada. Association for Computational Linguistics.

Omer Farooq Ahmed Adeeb and Seyed Jahanshah Kabudian. 2025. Steggpt: A novel foundation-model-based character-level linguistic steganography method utilizing large language models. *IEEE Access*, 13:89915–89925.

Yapei Feng, Feng Jiang, Shanhao Wu, and Hua Zhong. 2025. A high-capacity and secure disambiguation algorithm for neural linguistic steganography. *Preprint*, arXiv:2510.02332.

Jessica Fridrich and Miroslav Goljan. 2002. Practical steganalysis of digital images: state of the art. In *Security and Watermarking of Multimedia Contents IV*, volume 4675, pages 1 – 13. International Society for Optics and Photonics, SPIE.

Zhangjie Fu, Qi Yu, Fan Wang, and Changhao Ding. 2022. Hga: Hierarchical feature extraction with graph and attention mechanism for linguistic steganalysis. *IEEE Signal Processing Letters*, 29:1734–1738.

Kai Gao, Ji-Hwei Horng, Ching-Chun Chang, and Chin-Chen Chang. 2024a. Linguistic secret sharing via ambiguous token selection for iot security. *Electronics*, 13(21).

Liting Gao, Juan Wen, Ziwei Zhang, Guangying Fan, and Yinghan Zhou. 2024b. Linguistic steganalysis via probabilistic weighted contrastive learning. *IEEE Signal Processing Letters*, 31:1875–1879.

Joseph Gardiner. 2012. Stegchat: a synonym-substitution based algorithm for text steganography. *Submitted in conformity with the requirements for the degree of MSc Computer Security, School of Computer Science, University of Birmingham*, pages 1–63.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

Christian Grothoff, Krista Grothoff, Ludmila Alkhutova, Ryan Stutsman, and Mikhail Atallah. 2005. Translation-based steganography. In *Information Hiding*, pages 219–233, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A survey on llm-as-a-judge. *Preprint*, arXiv:2411.15594.

Shengnan Guo, Xuekai Chen, Zhuang Wang, Zhongliang Yang, and Linna Zhou. 2024. Linguistic steganalysis based on clustering and ensemble learning in imbalanced scenario. In *Digital Forensics and Watermarking*, pages 304–318, Singapore. Springer Nature Singapore.

Shengnan Guo, Jianyi Liu, Zhongliang Yang, Weike You, and Ru Zhang. 2022. Linguistic steganalysis merging semantic and statistical features. *IEEE Signal Processing Letters*, 29:2128–2132.

Yuzhe Guo, Zhongliang Yang, Zhuang Wang, Zhili Zhou, and Linna Zhou. 2025. Secc-stega: Generative linguistic steganographic framework based on error correcting codes. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

R Gurunath and Debabrata Samanta. 2023. A new 3-bit hiding covert channel algorithm for public data and medical data security using format-based text steganography. *Journal of Database Management (JDM)*, 34(2):1–22.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy. PMLR.

Te Sun Han. 2005. Folklore in source coding: information-spectrum approach. *IEEE Trans. Inf. Theor.*, 51(2):747–753.

Chris Honsinger. 2002. Digital Watermarking. *Journal of Electronic Imaging*, 11(3):414.

Nicholas J. Hopper, John Langford, and Luis von Ahn. 2002. Provably secure steganography. In *Advances in Cryptology — CRYPTO 2002*, pages 77–92, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sachin Hosmani, H. G. Rama Bhat, and K. Chandrasekaran. 2015. Dual stage text steganography using unicode homoglyphs. In *Security in Computing and Communications*, pages 265–276, Cham. Springer International Publishing.

Kaibo Huang, Zipei Zhang, Yukun Wei, TianXin Zhang, Zhongliang Yang, and Linna Zhou. 2025a. Gsd-fuse: Capturing cognitive inconsistencies from multi-dimensional weak signals in social media steganalysis. *Preprint*, arXiv:2505.17085.

Yu-Shin Huang, Peter Just, Krishna Narayanan, and Chao Tian. 2024. Od-stega: Llm-based near-imperceptible steganography via optimized distributions. *Preprint*, arXiv:2410.04328.

Yu-Shin Huang, Chao Tian, Krishna Narayanan, and Lizhong Zheng. 2025b. Relatively-secure llm-based steganography via constrained markov decision processes. *Preprint*, arXiv:2502.01827.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Sai-Mei Jiao, Hai-feng Wang, Kun Zhang, and Ya-qi Hu. 2021. Neural linguistic steganalysis via multi-head self-attention. *Journal of Electrical and Computer Engineering*, 2021(1):6668369.

Kaggle. 2017. All the news. `https://www.kaggle.com/snapcrack/all-the-news/data`. Accessed: 2017-08-20.

David Kahn. 1996. The history of steganography. In *Proceedings of the First International Workshop on Information Hiding*, page 1–5, Berlin, Heidelberg. Springer-Verlag.

Huixian Kang, Hanzhou Wu, and Xinpeng Zhang. 2020. Generative text steganography based on lstm network and attention mechanism with keywords. *Electronic Imaging*, 32(4):291–1–291–1.

Gabriel Kaptchuk, Tushar M. Jois, Matthew Green, and Aviel D. Rubin. 2021. Meteor: Cryptographically secure steganography for realistic distributions. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 1529–1548, New York, NY, USA. Association for Computing Machinery.

Stefan Katzenbeisser and Fabien AP Petitcolas. 2002. Defining security in steganographic systems. In *Security and Watermarking of Multimedia Contents IV*, volume 4675, pages 50–56. SPIE.

Katia Lida Kermanidis and Emmanouil Magkos. 2009. Empirical paraphrasing of modern greek text in two phases: An application to steganography. In *Computational Linguistics and Intelligent Text Processing*, pages 535–546, Berlin, Heidelberg. Springer Berlin Heidelberg.

Md. Khairullah. 2009. A novel text steganography system using font color of the invisible characters in microsoft word documents. In *2009 Second International Conference on Computer and Electrical Engineering*, volume 1, pages 482–484.

Rajeev Kumar, Satish Chand, and Samayveer Singh. 2015. An efficient text steganography scheme using unicode space characters. *Int. J. Forensic Comput. Sci*, 10(1):8–14.

Rajeev Kumar, Aruna Malik, Samayveer Singh, Bijendra Kumar, and Satish Chand. 2016. A space based reversible high capacity text steganography scheme using font type and style. In *2016 International Conference on Computing, Communication and Automation (ICCCA)*, pages 1090–1094.

Oleksandr Kuznetsov, Kyrylo Chernov, Aigul Shaikhanova, Kainizhamal Iklassova, and Dinara Kozhakhmetova. 2025. Deepstego: Privacy-preserving natural language steganography using large language models and advanced neural architectures. *Computers*, 14(5).

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Enlu Li, Zhangjie Fu, Siyu Chen, and Junfu Chen. 2020. A two-stage highly robust text steganalysis model. *Journal of Cybersecurity*, 2(4):183.

Fanxiao Li, Ping Wei, Tingchao Fu, Yu Lin, and Wei Zhou. 2024a. Imperceptible text steganography based on group chat. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

Fanxiao Li, Sixing Wu, Jiong Yu, Shuoxin Wang, Bing-Bing Song, Renyang Liu, Haoseng Lai, and Wei Zhou. 2023a. Rewriting-stego: Generating natural and controllable steganographic text with pre-trained language model. In *Database Systems for Advanced Applications*, pages 617–626, Cham. Springer Nature Switzerland.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Songbin Li, Hui Du, and Jingang Wang. 2025a. General steganalysis of generative linguistic steganography based on dynamic segment-level lexical association extraction. *IEEE Signal Processing Letters*, 32:191–195.

Songbin Li, Jingang Wang, and Peng Liu. 2023b. Detection of generative linguistic steganography based on explicit and latent text word relation mining using deep learning. *IEEE Transactions on Dependable and Secure Computing*, 20(2):1476–1487.

Xinxin Li, Zichi Wang, and Xinpeng Zhang. 2025b. Black-box steganography for large language models. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1.

Yamin Li, Jun Zhang, Zhongliang Yang, and Ru Zhang. 2021. Topic-aware neural linguistic steganography based on knowledge graphs. *ACM/IMS Trans. Data Sci.*, 2(2).

Yihao Li, Ru Zhang, Jianyi Liu, and Qi Lei. 2024b. A semantic controllable long text steganography framework based on llm prompt engineering and knowledge graph. *IEEE Signal Processing Letters*, 31:2610–2614.

Guorui Liao, Jinshuai Yang, Kaiyi Pang, and Yongfeng Huang. 2024. Co-stega: Collaborative linguistic steganography for the low capacity challenge in social media. In *Proceedings of the 2024 ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '24, page 7–12, New York, NY, USA. Association for Computing Machinery.

Guorui Liao, Jinshuai Yang, Weizhi Shao, and Yongfeng Huang. 2025. A framework for designing provably secure steganography. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 6837–6856.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chin-Yew Lin and Franz Josef Och. 2004. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 501–507, Geneva, Switzerland. COLING.

Ke Lin, Yiyang Luo, Zijian Zhang, and Luo Ping. 2024. Zero-shot generative linguistic steganography. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5168–5182, Mexico City, Mexico. Association for Computational Linguistics.

YiQing Lin and ZhongHua Wang. 2024. A novel method for linguistic steganography by english translation using attention mechanism and probability distribution theory. *Plos one*, 19(1):e0295207.

Yiting Liu, Chungen Xu, Fei Yang, Pan Zhang, and Linlong Wang. 2025. Linguistic steganography via self-adjusting asymmetric number system. *Computational Linguistics*, pages 1–35.

Yuling Liu, Cuilin Wang, Jie Wang, Bo Ou, and Xin Liao. 2024. Generative steganography via live comments on streaming video frames. *IEEE Transactions on Computational Social Systems*, 11(3):4412–4425.

Yunxia Liu, Shuyang Liu, Yonghao Wang, Hongguo Zhao, and Si Liu. 2019. Video steganography: A review. *Neurocomputing*, 335:238–250.

Yilin Long, Zhongliang Yang, Zhuang Wang, Zhili Zhou, Yongfeng Huang, and Linna Zhou. 2025. Scf-stega: Controllable linguistic steganography based on semantic communications framework. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Junkai Lu, Zhongliang Yang, Kaibo Huang, Zhuang Wang, Zhili Zhou, and Linna Zhou. 2025. Tgca: A transformer gnn-based approach with cross-attention mechanism for steganographic text detection in social networks. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Tianhe Lu, Gongshen Liu, Ru Zhang, and Tianjie Ju. 2023a. Neural linguistic steganography with controllable security. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Tianhe Lu, Gongshen Liu, Ru Zhang, Peixuan Li, and Tianjie Ju. 2023b. Robust secret data hiding for transformer-based neural machine translation. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Yubo Luo and Yongfeng Huang. 2017. Text steganography with high embedding rate: Using recurrent neural networks to generate chinese classic poetry. In *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '17, page 99–104, New York, NY, USA. Association for Computing Machinery.

Yubo Luo, Yongfeng Huang, Fufang Li, and Chinchen Chang. 2016. Text steganography based on ci-poetry generation using markov chain model. *KSII Trans. Internet Inf. Syst.*, 10:4568–4584.

Yufei Luo, Zhen Yang, Xin Xu, Yelei Wang, Ru Zhang, and Yongfeng Huang. 2025a. Clustering-driven pseudo-labeling in source-free domain adaptation for linguistic steganalysis. In *Proceedings of the 2025 ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSEC '25, page 153–163, New York, NY, USA. Association for Computing Machinery.

Yufei Luo, Zhen Yang, Ru Zhang, and Jianyi Liu. 2025b. Pseudo-label based domain adaptation for zero-shot text steganalysis. In *Computational and Experimental Simulations in Engineering*, pages 128–142, Cham. Springer Nature Switzerland.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Susmita Mahato, Danish Ali Khan, and Dilip Kumar Yadav. 2020. A modified approach to data hiding in microsoft word documents by change-tracking technique. *Journal of King Saud University - Computer and Information Sciences*, 32(2):216–224.

Susmita Mahato, Dilip Kumar Yadav, and Danish Ali Khan. 2014. A novel approach to text steganography using font size of invisible space characters in microsoft word document. In *Intelligent Computing, Networking, and Informatics*, pages 1047–1054, New Delhi. Springer India.

Mohammed Abdul Majeed, Rossilawati Sulaiman, Zarina Shukur, and Mohammad Kamrul Hasan. 2021. A review on text steganography techniques. *Mathematics*, 9(21).

Ralph C. Merkle. 1978. Secure communications over insecure channels. *Commun. ACM*, 21(4):294–299.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

George A. Miller. 1995. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41.

Rina Mishra and Praveen Bhanodiya. 2015. A review on steganography and cryptography. In *2015 International Conference on Advances in Computer Engineering and Applications*, pages 119–122.

H. Hernan Moraldo. 2014. An approach for text steganography based on markov chains. *Preprint*, arXiv:1409.0915.

Hasanah Zulcefli Muhammad, Sharifah Mumtazah Syed Ahmad Abdul Rahman, and Asma Shakil. 2009. Synonym based malay linguistic text steganography. In *2009 Innovative Technologies in Intelligent Systems and Industrial Applications*, pages 423–427.

Alfonso Muñoz, Justo Carracedo Gallardo, and Irina Argüelles Álvarez. 2010. Improving n-gram linguistic steganography based on templates. In *2010 International Conference on Security and Cryptography (SECRYPT)*, pages 1–4.

Alfin Naharuddin, Adhi Dharma Wibawa, and Surya Sumpeno. 2018. A high capacity and imperceptible text steganography using binary digit mapping on ascii characters. In *2018 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pages 287–292.

M. V. Namitha, G. R. Manjula, and Manjula C. Belavagi. 2024. Stegabb: A cover-generating text steganographic tool using gpt-3 language modeling for covert communication across sdrs. *IEEE Access*, 12:82057–82067.

Arooj Nissar and A.H. Mir. 2010. Classification of steganalysis techniques: A study. *Digital Signal Processing*, 20(6):1758–1770.

Qingying Niu and Zhen Yang. 2025. Text steganalysis with language style for social. *Computational and Experimental Simulations in Engineering: Proceedings of ICCES 2024—Volume 4*, 176:116.

Yan Niu, Juan Wen, Ping Zhong, and Yiming Xue. 2019. A hybrid r-bilstm-c neural network based text steganalysis. *IEEE Signal Processing Letters*, 26(12):1907–1911.

Jumon Nozaki and Yugo Murawaki. 2022. Addressing segmentation ambiguity in neural linguistic steganography. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 109–116, Online only. Association for Computational Linguistics.

Open Source Initiative. 2024. The open source AI definition 1.0. `https://opensource.org/ai/open-source-ai-definition`.

Chengfu Ou, Lingyun Xiang, and Yangfan Liu. 2025. Promising multi-granularity linguistic steganography by jointing syntactic and lexical manipulations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(23):24984–24992.

Chao Pan, Donghui Hu, Yaofei Wang, Kejiang Chen, Yinyin Peng, Xianjin Rong, Chen Gu, and Meng Li. 2025. Rethinking prefix-based steganography for enhanced security and efficiency. *IEEE Transactions on Information Forensics and Security*, 20:3287–3301.

Kaiyi Pang. 2025a. Frestega: A plug-and-play method for boosting imperceptibility and capacity in generative linguistic steganography for real-world scenarios. *Preprint*, arXiv:2412.19652.

Kaiyi Pang. 2025b. Provable secure steganography based on adaptive dynamic sampling. *Preprint*, arXiv:2504.12579.

Kaiyi Pang, Minhao Bai, Jinshuai Yang, Huili Wang, Minghu Jiang, and Yongfeng Huang. 2024a. Fremax: A simple method towards truly secure generative linguistic steganography. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4755–4759.

Kaiyi Pang, Minhao Bai, Jinshuai Yang, Wei-Qiang Zhang, Minghu Jiang, and Yongfeng Huang. 2025. Winstega: An adaptive robust enhancement framework for generative linguistic steganography. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Kaiyi Pang, Jinshuai Yang, Yue Gao, Minhao Bai, Zhongliang Yang, Minghu Jiang, and Yongfeng Huang. 2024b. Cats: Connection-aware and interaction-based text steganalysis in social networks. In *Neural Information Processing*, pages 109–121, Singapore. Springer Nature Singapore.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Wanli Peng, Sheng Li, Zhenxing Qian, and Xinpeng Zhang. 2023a. Text steganalysis based on hierarchical supervised learning and dual attention mechanism. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:3513–3526.

Wanli Peng, Tao Wang, Zhenxing Qian, Sheng Li, and Xinpeng Zhang. 2023b. Cross-modal text steganography against synonym substitution-based text attack. *IEEE Signal Processing Letters*, 30:299–303.

Wanli Peng, Jinyu Zhang, Yiming Xue, and Zhenghong Yang. 2021. Real-time text steganalysis based on multi-stage transfer learning. *IEEE Signal Processing Letters*, 28:1510–1514.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Neil Perry, Sanket Gupte, Nishant Pitta, and Lior Rotem. 2025. Robust steganography from large language models. *Preprint*, arXiv:2504.08977.

Lip Yee Por, KokSheik Wong, and Kok Onn Chee. 2012. Unispach: A text-based data hiding method using unicode space characters. *Journal of Systems and Software*, 85(5):1075–1082.

Cao Qi, Sun Xingming, and Xiang Lingyun. 2013. A secure text steganography based on synonym substitution. In *IEEE Conference Anthology*, pages 1–3.

Yuang Qi, Kejiang Chen, Kai Zeng, Weiming Zhang, and Nenghai Yu. 2025a. Provably secure disambiguating neural linguistic steganography. *IEEE Transactions on Dependable and Secure Computing*, 22(3):2430–2442.

Yuang Qi, Na Zhao, Qiyi Yao, Benlong Wu, Weiming Zhang, Nenghai Yu, and Kejiang Chen. 2025b. STEAD: Robust provably secure linguistic steganography with diffusion language model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zezheng Qin, Congcong Sun, Taiyi He, Yuke He, Azizol Abdullah, Normalia Samian, and Nuur Alifah Roslan. 2024. Adlm – stega: A universal adaptive token selection algorithm for improving steganographic text quality via information entropy. *Preprint*, arXiv:2410.20825.

Feno Heriniaina Rabevohitra and Yantao Li. 2019. Text cover steganography using font color of the invisible characters and optimized primary color-intensities. In *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, pages 1704–1708.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Khan Farhan Rafat. 2009. Enhanced text steganography by changing word's spelling. In *Proceedings of the 7th International Conference on Frontiers of Information Technology*, FIT '09, New York, NY, USA. Association for Computing Machinery.

Abdul Monem S Rahma, Wesam S Bhaya, and Dhamyaa A Al-Nasrawi. 2013. Text steganography based on unicode of characters in multilingual. *International Journal of Engineering Research and Applications*, 3(4):1153–1165.

Bala Krishnan Ramakrishnan, Prasanth Kumar Thandra, and AV Satya Murty Srinivasula. 2016. Text steganography: a novel character-level embedding algorithm using font attribute. *Security and Communication Networks*, 9(18):6066–6079.

Ram Sundara Raman, Adrian Stoll, Jakub Dalek, Reethika Ramesh, Will Scott, and Roya Ensafi. 2020. Measuring the deployment of network censorship filters at global scale. In *NDSS*.

Ronald L. RIVEST. 1990. Chapter 13 - cryptography. In JAN VAN LEEUWEN, editor, *Algorithms and Complexity*, Handbook of Theoretical Computer Science, pages 717–755. Elsevier, Amsterdam.

Esra Satir and Hakan Isik. 2012. A compression-based text steganography method. *Journal of Systems and Software*, 85(10):2385–2394. Automated Software Evolution.

Elsa Serret, Antoine Lesueur, and Alban Gabillon. 2022. Linguistic steganography for messaging applications. In *ICISSP*, pages 519–527.

Jiaming Shen, Heng Ji, and Jiawei Han. 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)*, pages 303–313, Online. Association for Computational Linguistics.

Hao Shi, Wenpu Guo, and Shaoyuan Gao. 2025. Emotionally controllable text steganography based on large language model and named entity. *Technologies*, 13(7).

M. Hassan Shirali-Shahreza and Mohammad Shirali-Shahreza. 2008. A new synonym text steganography. In *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 1524–1526.

Mohammad Shirali-Shahreza. 2008. Text steganography by changing words spelling. In *2008 10th International Conference on Advanced Communication Technology*, volume 3, pages 1912–1913.

Alexey Nikolaevich Shniperov and KA Nikitina. 2016. A text steganography method based on markov chains. *Automatic Control and Computer Sciences*, 50(8):802–808.

Gustavus J. Simmons. 1984. *The Prisoners' Problem and the Subliminal Channel*, pages 51–67. Springer US, Boston, MA.

Prem Singh, Rajat Chaudhary, and Ambika Agarwal. 2012. A novel approach of text steganography based on null spaces. *IOSR Journal of Computer Engineering*, 3(4):11–17.

Martin Steinebach. 2024. Natural language steganography by chatgpt. In *Proceedings of the 19th International Conference on Availability, Reliability and Security*, ARES '24, New York, NY, USA. Association for Computing Machinery.

Ryan Stutsman, Christian Grothoff, Mikhail Atallah, and Krista Grothoff. 2006. Lost in just the translation. In *Proceedings of the 2006 ACM Symposium on Applied Computing*, SAC '06, page 338–345, New York, NY, USA. Association for Computing Machinery.

Nandhini Subramanian, Omar Elharrouss, Somaya Al-Maadeed, and Ahmed Bouridane. 2021. Image steganography: A review of the recent advances. *IEEE Access*, 9:23409–23423.

Bowen Sun, Yamin Li, Jun Zhang, Honghong Xu, Xiaoqiang Ma, and Ping Xia. 2023. Topic controlled steganography via graph-to-text generation. *CMES - Computer Modeling in Engineering and Sciences*, 136(1):157–176.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *Preprint*, arXiv:1904.09223.

Roger J Sutton. 2002. *Secure communications: Applications and management*. John Wiley & Sons.

P.F. Syverson, D.M. Goldschlag, and M.G. Reed. 1997. Anonymous connections and onion routing. In *Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No.97CB36097)*, pages 44–54.

Milad Taleby Ahvanooey, Qianmu Li, Jun Hou, Hassan Dana Mazraeh, and Jing Zhang. 2018. Aitsteg: An innovative text steganography technique for hidden transmission of text message via social media. *IEEE Access*, 6:65981–65995.

Yifan Tang, Yihao Wang, Ru Zhang, and Jianyi Liu. 2025. Linguistic steganalysis via llms: Two modes for efficient detection of strongly concealed stego. *IEEE Signal Processing Letters*, 32:541–545.

Yongju Tong, YuLing Liu, Jie Wang, and Guojiang Xin. 2019. Text steganography on rnn-generated lyrics. *Mathematical Biosciences and Engineering*, 16(5):5451–5463.

Don J. Torrieri. 1992. *Principles of Secure Communication Systems*, 2nd edition. Artech House, Inc., USA.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Honai Ueoka, Yugo Murawaki, and Sadao Kurohashi. 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*, pages 5486–5492, Online. Association for Computational Linguistics.

Merve Varol Arısoy. 2022. Lzw-cie: a high-capacity linguistic steganography based on lzw char index encoding. *Neural Computing and Applications*, 34(21):19117–19145.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ei Nyein Chan Wai and May Aye Khine. 2011. Syntactic bank-based linguistic steganography approach. In *2011 International Conference on Information Communication and Management IPCSIT*, volume 16.

Cuilin Wang, Yuling Liu, Yongju Tong, and Jingwen Wang. 2021. Gan-gls: Generative lyric steganography based on generative adversarial networks. *Computers, Materials and Continua*, 69(1):1375–1390.

Huili Wang, Zhongliang Yang, Jinshuai Yang, Cheng Chen, and Yongfeng Huang. 2023a. Linguistic steganalysis in few-shot scenario. *IEEE Transactions on Information Forensics and Security*, 18:4870–4882.

Huili Wang, Zhongliang Yang, Jinshuai Yang, Yue Gao, and Yongfeng Huang. 2024a. Hi-stega: A hierarchical linguistic steganography framework combining retrieval and generation. In *Neural Information Processing*, pages 41–54, Singapore. Springer Nature Singapore.

Jingang Wang, Songbin Li, and Hui Du. 2024b. Cross-domain detection of generative linguistic steganography based on source and target domain in-depth alignment. In *2024 4th International Conference on Blockchain Technology and Information Security (ICBCTIS)*, pages 124–128.

Qiyan Wang, Xun Gong, Giang T.K. Nguyen, Amir Houmansadr, and Nikita Borisov. 2012. Censor-spoofer: asymmetric communication using ip spoofing for censorship-resistant web browsing. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, page 121–132, New York, NY, USA. Association for Computing Machinery.

Rong Wang, Lingyun Xiang, Yangfan Liu, and Chunfang Yang. 2023b. Png-stega: Progressive non-autoregressive generative linguistic steganography. *IEEE Signal Processing Letters*, 30:528–532.

Shuoxin Wang, Fanxiao Li, Jiong Yu, Haosen Lai, Sixing Wu, and Wei Zhou. 2023c. Enhancing semantic consistency in linguistic steganography via denosing auto-encoder and semantic-constrained huffman coding. In *Natural Language Processing and Chinese Computing*, pages 799–812, Cham. Springer Nature Switzerland.

Xilong Wang, Yaofei Wang, Kejiang Chen, Jinyang Ding, Weiming Zhang, and Nenghai Yu. 2023d. Ic-stega: Image captioning-based semantically controllable linguistic steganography. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Yaofei Wang, Gang Pei, Kejiang Chen, Jinyang Ding, Chao Pan, Weilong Pang, Donghui Hu, and Weiming

Zhang. 2025a. SparSamp: Efficient provably secure steganography based on sparse sampling. In *The 34th USENIX Security Symposium*.

Yihao Wang, Ruiqi Song, Lingxiao Li, Yifan Tang, Ru Zhang, and Jianyi Liu. 2026. User profile constructed by multiple attributes for optimizing linguistic steganalysis in social networks. *Expert Systems with Applications*, 297:129311.

Yihao Wang, Ruiqi Song, Lingxiao Li, Ru Zhang, and Jianyi Liu. 2025b. Dynamically allocated interval-based generative linguistic steganography with roulette wheel. *Applied Soft Computing*, 176:113101.

Yihao Wang, Ru Zhang, and Jianyi Liu. 2023e. Rls-dts: Reinforcement-learning linguistic steganalysis in distribution-transformed scenario. *IEEE Signal Processing Letters*, 30:1232–1236.

Yihao Wang, Ru Zhang, and Jianyi Liu. 2023f. V-a3ts: A rapid text steganalysis method based on position information and variable parameter multi-head self-attention controlled by length. *Journal of Information Security and Applications*, 75:103512.

Zhuang Wang, Xuekai Chen, Zhongliang Yang, and Linna Zhou. 2025c. Kike: Linguistic steganalysis based on knowledge infusion and knowledge encoding. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Zhuang Wang, Linna Zhou, Xuekai Chen, Zhili Zhou, and Zhongliang Yang. 2025d. Stlc-kg:a social text steganalysis method combining large-scale language models and common-sense knowledge graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(24):25461–25469.

Peter Wayner. 1992. Mimic functions. *Cryptologia*, 16(3):193–214.

Juan Wen, Yaqian Deng, Wanli Peng, and Yiming Xue. 2023a. Linguistic steganalysis via fusing multi-granularity attentional text features. *Chinese Journal of Electronics*, 32(1):76–84.

Juan Wen, Yaqian Deng, Jiaxuan Wu, Xingpeng Liu, and Yiming Xue. 2022a. Lifelong learning for text steganalysis based on chronological task sequence. *IEEE Signal Processing Letters*, 29:2412–2416.

Juan Wen, Liting Gao, Guangying Fan, Ziwei Zhang, Jianghao Jia, and Yiming Xue. 2023b. Scl-stega: Exploring advanced objective in linguistic steganalysis using contrastive learning. In *Proceedings of the 2023 ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '23, page 97–102, New York, NY, USA. Association for Computing Machinery.

Juan Wen, Ziwei Zhang, Yu Yang, and Yiming Xue. 2022b. Few-shot text steganalysis based on attentional meta-learner. In *Proceedings of the 2022 ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '22, page 97–106, New York, NY, USA. Association for Computing Machinery.

Juan Wen, Xuejing Zhou, Ping Zhong, and Yiming Xue. 2019. Convolutional neural network based text steganalysis. *IEEE Signal Processing Letters*, 26(3):460–464.

Matt White, Ibrahim Haddad, Cailean Osborne, Xiao-Yang Yanglet Liu, Keyi Wang, Ahmed Abdelmonsef, Sachin Mathew Varghese, and Arnaud Le Hors. 2025. The model openness framework: Promoting completeness and openness for reproducibility, transparency, and usability in artificial intelligence. In *Proceedings of the Workshop on Regulatable ML at NeurIPS 2025*.

Alex Wilson, Phil Blunsom, and Andrew D Ker. 2014. Linguistic steganography on twitter: hierarchical language modeling with manual interaction. In *Media Watermarking, Security, and Forensics 2014*, volume 9028, pages 9–25. SPIE.

Alex Wilson and Andrew D Ker. 2016. Avoiding detection on twitter: embedding strategies for linguistic steganography. *Electronic Imaging*, 28:1–9.

Kamil Woźniak, Marek R. Ogiela, and Lidia Ogiela. 2025. Multi-criteria linguistic optimization for covert communication in secure llm-based steganography. *Applied Soft Computing*, 185:113960.

Hanzhou Wu, Tianyu Yang, Xiaoyan Zheng, and Yurun Fang. 2024a. *Linguistic Steganography and Linguistic Steganalysis*, pages 163–190. Springer Nature Switzerland, Cham.

Hanzhou Wu, Biao Yi, Feng Ding, Guorui Feng, and Xinpeng Zhang. 2021. Linguistic steganalysis with graph neural networks. *IEEE Signal Processing Letters*, 28:558–562.

Jiaxuan Wu, Xin Chen, Juan Wen, Wanli Peng, and Yiming Xue. 2024b. A test-time entropy minimization method for cross-domain linguistic steganalysis. *IEEE Signal Processing Letters*, 31:1760–1764.

Jiaxuan Wu, Zhengxian Wu, Yiming Xue, Juan Wen, and Wanli Peng. 2024c. Generative text steganography with large language model. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, page 10345–10353, New York, NY, USA. Association for Computing Machinery.

Ning Wu, Zhongliang Yang, Yi Yang, Lian Li, Poli Shang, Weibo Ma, and Zhenru Liu. 2020. Stbs-stega: Coverless text steganography based on state transition-binary sequence. *International Journal of Distributed Sensor Networks*, 16(3):1550147720914257.

Zhengxian Wu, Juan Wen, Yiming Xue, Ziwei Zhang, and Yinghan Zhou. 2025. Gtsd: Generative text steganography based on diffusion model. In *Neural Information Processing*, pages 168–183, Singapore. Springer Nature Singapore.

Lingyun Xiang, Xu He, Xi Zhang, and Chengfu Ou. 2025. Linguistic steganography based on sentence attribute encoding. *Computers, Materials and Continua*, 84(2):2375–2389.

Lingyun Xiang, Yuhang Liu, Huiqing You, and Chengfu Ou. 2022a. Aggregating local and global text features for linguistic steganalysis. *IEEE Signal Processing Letters*, 29:1502–1506.

Lingyun Xiang, Chengfu Ou, and Daojian Zeng. 2024. Linguistic steganography: Hiding information in syntax space. *IEEE Signal Processing Letters*, 31:261–265.

Lingyun Xiang, Xingming Sun, Gang Luo, and Can Gan. 2007. Research on steganalysis for text steganography based on font format. In *Third International Symposium on Information Assurance and Security*, pages 490–495.

Lingyun Xiang, Rong Wang, Yuhang Liu, Yangfan Liu, and Lina Tan. 2023a. A general linguistic steganalysis framework using multi-task learning. *Computer Systems Science & Engineering*, 46(2).

Lingyun Xiang, Rong Wang, Zhongliang Yang, and Yuling Liu. 2022b. Generative linguistic steganography: A comprehensive review. *KSII Transactions on Internet & Information Systems*, 16(3).

Lingyun Xiang, Xinhui Wang, Chunfang Yang, and Peng Liu. 2017. A novel linguistic steganography based on synonym run-length encoding. *IEICE transactions on Information and Systems*, 100(2):313–322.

Lingyun Xiang, Jiali Xia, Yangfan Liu, and Yan Gui. 2023b. Cpg-ls: Causal perception guided linguistic steganography. *IEEE Signal Processing Letters*, 30:1762–1766.

Lingyun Xiang, Shuanghui Yang, Yuhang Liu, Qian Li, and Chengzhang Zhu. 2020. Novel linguistic steganography based on character-level text generation. *Mathematics*, 8(9).

Lingyun Xiang, Xiao Yang, Jiahe Zhang, and Weizheng Wang. 2019. A word-frequency-preserving steganographic method based on synonym substitution. *International Journal of Computational Science and Engineering*, 19(1):132–139.

Tong Xiao, Jingang Wang, and Songbin Li. 2025. Heterogeneous domain remapping for universal detection of generative linguistic steganography. *IEEE Signal Processing Letters*, 32:1281–1285.

Qiong Xu, Ru Zhang, and Jianyi Liu. 2023. Linguistic steganalysis by enhancing and integrating local and global features. *IEEE Signal Processing Letters*, 30:16–20.

Yimin Xu, Liran Yang, Tengyun Zhao, and Ping Zhong. 2021. A novel linguistic steganalysis method for hybrid steganographic texts. *Journal of Physics: Conference Series*, 1873(1):012053.

Yimin Xu, Tengyun Zhao, and Ping Zhong. 2022. Small-scale linguistic steganalysis for multi-concealed scenarios. *IEEE Signal Processing Letters*, 29:130–134.

Yiming Xue, Lingzhi Kong, Wanli Peng, Ping Zhong, and Juan Wen. 2022a. An effective linguistic steganalysis framework based on hierarchical mutual learning. *Information Sciences*, 586:140–154.

Yiming Xue, Jiaxuan Wu, Ronghua Ji, Ping Zhong, Juan Wen, and Wanli Peng. 2024. Adaptive domain-invariant feature extraction for cross-domain linguistic steganalysis. *IEEE Transactions on Information Forensics and Security*, 19:920–933.

Yiming Xue, Boya Yang, Yaqian Deng, Wanli Peng, and Juan Wen. 2022b. Domain adaptational text steganalysis based on transductive learning. In *Proceedings of the 2022 ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '22, page 91–96, New York, NY, USA. Association for Computing Machinery.

Ruiyi Yan and Yugo Murawaki. 2025. Addressing tokenization inconsistency in steganography and watermarking based on large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7087–7109, Suzhou, China. Association for Computational Linguistics.

Ruiyi Yan, Tian Song, and Yating Yang. 2024a. A near-imperceptible disambiguating approach via verification for generative linguistic steganography. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1638–1643.

Ruiyi Yan, Tian Song, and Yating Yang. 2024b. Tokenfree: A tokenization-free generative linguistic steganographic approach with enhanced imperceptibility. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 449–455.

Ruiyi Yan, Yating Yang, and Tian Song. 2023. A secure and disambiguating approach for generative linguistic steganography. *IEEE Signal Processing Letters*, 30:1047–1051.

Boya Yang, Wanli Peng, Yiming Xue, and Ping Zhong. 2021a. A generation-based text steganography by maintaining consistency of probability distribution. *KSII Trans. Internet Inf. Syst.*, 15(11):4184–4202.

Hao Yang, YongJian Bao, Zhongliang Yang, Sheng Liu, Yongfeng Huang, and Saimei Jiao. 2020a. Linguistic steganalysis via densely connected lstm with feature pyramid. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '20, page 5–10, New York, NY, USA. Association for Computing Machinery.

Jinshuai Yang, Zhongliang Yang, Xinrui Ge, Jiajun Zou, Yue Gao, and Yongfeng Huang. 2023a. Link: Linguistic steganalysis framework with external knowledge. In *ICASSP 2023 - 2023 IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5.

Jinshuai Yang, Zhongliang Yang, Siyu Zhang, Haoqin Tu, and Yongfeng Huang. 2022a. Sesy: Linguistic steganalysis framework integrating semantic and syntactic features. *IEEE Signal Processing Letters*, 29:31–35.

Jinshuai Yang, Zhongliang Yang, Jiajun Zou, Haoqin Tu, and Yongfeng Huang. 2023b. Linguistic steganalysis toward social network. *IEEE Transactions on Information Forensics and Security*, 18:859–871.

Minhao Bai. Jinshuai Yang, Kaiyi Pang, Huili Wang, and Yongfeng Huang. 2024a. Towards next-generation steganalysis: Llms unleash the power of detecting steganography. *Preprint*, arXiv:2405.09090.

Rui Yang and Zhen-Hua Ling. 2019. Linguistic steganography by sampling-based language generation. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1014–1019.

Tianyu Yang, Hanzhou Wu, Biao Yi, Guorui Feng, and Xinpeng Zhang. 2024b. Semantic-preserving linguistic steganography by pivot translation and semantic-aware bins coding. *IEEE Transactions on Dependable and Secure Computing*, 21(1):139–152.

Zhen Yang, Zihan Xu, Xin Xu, Zhongliang Yang, and Ru Zhang. 2025. Linguistic steganalysis via text dual attention fusing statistical and multi-layer semantic features. *IEEE Signal Processing Letters*, 32:1371–1375.

Zhen Yang, Zihan Xu, Ru Zhang, and Yongfeng Huang. 2022b. T-gru: contextual gated recurrent unit model for high quality linguistic steganography. In *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6.

Zhong-Liang Yang, Xiao-Qing Guo, Zi-Ming Chen, Yong-Feng Huang, and Yu-Jin Zhang. 2019a. Rnn-stega: Linguistic steganography based on recurrent neural networks. *IEEE Transactions on Information Forensics and Security*, 14(5):1280–1295.

Zhong-Liang Yang, Si-Yu Zhang, Yu-Ting Hu, Zhi-Wen Hu, and Yong-Feng Huang. 2021b. Vae-stega: Linguistic steganography based on variational auto-encoder. *IEEE Transactions on Information Forensics and Security*, 16:880–895.

Zhongliang Yang, Baitao Gong, Yamin Li, Jinshuai Yang, Zhiwen Hu, and Yongfeng Huang. 2020b. Graph-stega: Semantic controllable steganographic text generation guided by knowledge graph. *Preprint*, arXiv:2006.08339.

Zhongliang Yang, Yongfeng Huang, and Yu-Jin Zhang. 2019b. A fast and efficient text steganalysis method. *IEEE Signal Processing Letters*, 26(4):627–631.

Zhongliang Yang, Yongfeng Huang, and Yu-Jin Zhang. 2020c. Ts-csw: text steganalysis and hidden capacity estimation based on convolutional sliding windows. *Multimedia Tools Appl.*, 79(25–26):18293–18316.

Zhongliang Yang, Shuyu Jin, Yongfeng Huang, Yujin Zhang, and Hui Li. 2018. Automatically generate steganographic text based on markov model and huffman coding. *Preprint*, arXiv:1811.04720.

Zhongliang Yang, Ke Wang, Jian Li, Yongfeng Huang, and Yu-Jin Zhang. 2019c. Ts-rnn: Text steganalysis based on recurrent neural networks. *IEEE Signal Processing Letters*, 26(12):1743–1747.

Zhongliang Yang, Nan Wei, Qinghe Liu, Yongfeng Huang, and Yujin Zhang. 2020d. Gan-tstega: Text steganography based on generative adversarial networks. In *Digital Forensics and Watermarking*, pages 18–31, Cham. Springer International Publishing.

Zhongliang Yang, Lingyun Xiang, Siyu Zhang, Xingming Sun, and Yongfeng Huang. 2021c. Linguistic generative steganography with enhanced cognitive-imperceptibility. *IEEE Signal Processing Letters*, 28:409–413.

Biao Yi, Hanzhou Wu, Guorui Feng, and Xinpeng Zhang. 2022a. Alisa: Acrostic linguistic steganography based on bert and gibbs sampling. *IEEE Signal Processing Letters*, 29:687–691.

Biao Yi, Hanzhou Wu, Guorui Feng, and Xinpeng Zhang. 2022b. Exploiting language model for efficient linguistic steganalysis. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3074–3078.

Huiqing You, Lingyun Xiang, Chunfang Yang, and Xiaobo Shen. 2024. Linguistic steganalysis via multi-task with crossing generative-natural domain. *Neurocomputing*, 603:128260.

Long Yu, Yuliang Lu, Xuehu Yan, and Yongqiang Yu. 2022. Mts-stega: Linguistic steganography based on multi-time-step. *Entropy*, 24(5):585.

ZS Yu, LS Huang, ZL Chen, L Li, W Yang, and XX Zhao. 2009. High embedding ratio text steganography by ci-poetry of the song dynasty. *Journal of Chinese Information Processing*, 23(4):55–62.

Kaiguo Yuan, Yu Yang, Ziwei Zhang, and Juan Wen. 2024. Multi-task few-shot text steganalysis based on context-attentive prototypes. *Expert Systems with Applications*, 249:123437.

Qingwei Zeng and Kaixi Wang. 2025. Steganographic text generation based on large language models in dialogue scenarios. In *Natural Language Processing and Chinese Computing*, pages 475–487, Singapore. Springer Nature Singapore.

Ruifan Zhang, Jianyi Liu, and Ru Zhang. 2024. Controllable semantic linguistic steganography via summarization generation. In *ICASSP 2024 - 2024 IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4560–4564.

Siyu Zhang, Zhongliang Yang, Jinshuai Yang, and Yongfeng Huang. 2021a. Linguistic steganography: From symbolic space to semantic space. *IEEE Signal Processing Letters*, 28:11–15.

Siyu Zhang, Zhongliang Yang, Jinshuai Yang, and Yongfeng Huang. 2021b. Provably secure generative linguistic steganography. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3046–3055, Online. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

Xin Zhang, Kejiang Chen, Na Zhao, Weiming Zhang, and Nenghai Yu. 2025a. Provably secure public-key steganography based on admissible encoding. *IEEE Transactions on Information Forensics and Security*, 20:3161–3175.

Ziwei Zhang, Juan Wen, Liting Gao, Wanli Peng, and Yiming Xue. 2025b. Linguistic steganalysis based on few-shot adversarial training. *IEEE Transactions on Dependable and Secure Computing*, 22(4):3514–3528.

Ziwei Zhang, Juan Wen, Wanli Peng, and Haowei Chang. 2025c. Student-augmented self-training with closed loop feedback in linguistic steganalysis. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, page 5515–5519, New York, NY, USA. Association for Computing Machinery.

Xiaoyan Zheng, Yurun Fang, and Hanzhou Wu. 2022. General framework for reversible data hiding in texts based on masked language modeling. In *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6.

Xiaoyan Zheng and Hanzhou Wu. 2022. Autoregressive linguistic steganography based on bert and consistency coding. *Security and Communication Networks*, 2022(1):9092785.

Jiuan Zhou, Yu Cheng, Yuan Xie, and Zhaoxia Yin. 2025. Auto-stega: An agent-driven system for lifelong strategy evolution in llm-based text steganography. *Preprint*, arXiv:2510.06565.

Xuejing Zhou, Wanli Peng, Boya Yang, Juan Wen, Yiming Xue, and Ping Zhong. 2022. Linguistic steganography based on adaptive probability distribution. *IEEE Transactions on Dependable and Secure Computing*, 19(5):2982–2997.

Zachary Ziegler, Yuntian Deng, and Alexander Rush. 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)*, pages 1210–1215, Hong Kong, China. Association for Computational Linguistics.

Jiajun Zou, Zhongliang Yang, Siyu Zhang, Sadaqat ur Rehman, and Yongfeng Huang. 2021. High-performance linguistic steganalysis, capacity estimation and steganographic positioning. In *Digital Forensics and Watermarking*, pages 80–93, Cham. Springer International Publishing.

Emır Öztürk, Andaç Şahın Mesut, and Özlem Aydin Fıdan. 2024. A character based steganography using masked language modeling. *IEEE Access*, 12:14248–14259.

A Comparison with Related Concepts

(1) **Cryptography** protects the content of a message, ensuring confidentiality, integrity, and authenticity, by transforming plaintext into ciphertext using mathematical algorithms and keys (RIVEST, 1990).

(2) **Watermarking** embeds information into a carrier to prove ownership, authenticity, or traceability, by inserting robust marks or patterns that survive common transformations (e.g., compression, distortion) (Honsinger, 2002).

In summary, Table 3 compares the features of steganography, cryptography, and watermarking. In addition, Figure 10 illustrates their procedures, respectively.

B Two Definitions of Steganographic Security

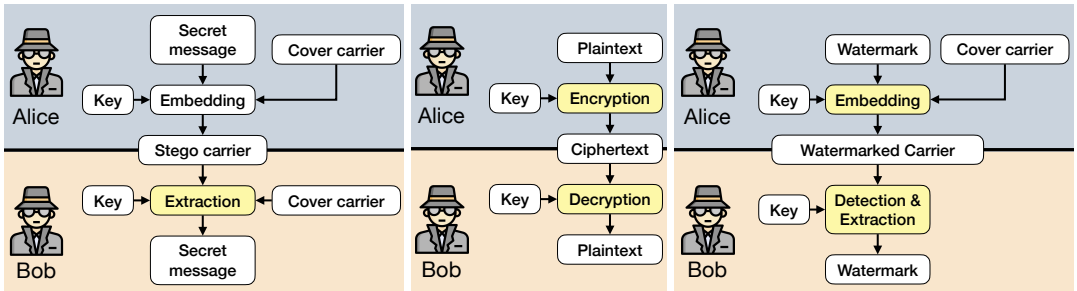
(1) Cachin (1998) was the first to model steganographic security from an **information-theoretic** perspective. Given an observation $\mathbf{x}$, the security of a steganographic system can be quantified by the Kullback–Leibler (KL) divergence between the cover distribution P_c and the stego distribution P_s:

$$D_{\mathrm{KL}}(P_c||P_s) = \sum_{\mathbf{x}\in\mathcal{D}} P_c(\mathbf{x}) \log \frac{P_c(\mathbf{x})}{P_s(\mathbf{x})}, \quad (1)$$

which measures the statistical difference between the two distributions. When $D_{\mathrm{KL}}(P_c||P_s) = 0$, the steganographic system is considered to be *perfectly secure*, as $P_c = P_s$ and thus the steganalyzer cannot distinguish the stego object from the cover object better than by random guessing.

Feature	Steganography	Cryptography	Watermarking
Goal	Hide the existence of communication	Protect the content of communication	Assert ownership or authenticity
Carrier	Multimedia	Plaintext → ciphertext	Multimedia
Visibility	Definitely not (ideally undetectable)	Obvious (ciphertext is visible)	Optional (depending on the watermarking type)
Adversary	Detect whether a given object contains hidden information	Recover the plaintext or secret key from the ciphertext	Destroy, alter, or remove the embedded watermark without significantly degrading the carrier quality
Robustness	Important (but optional in some work)	Strong against brute force	Designed to withstand distortion
Uses	Covert messaging, censorship evasion	Secure communication, authentication	Copyright protection, forensic tracking, digital rights management (DRM)

Table 3: Comparison of steganography, cryptography, and watermarking.



(a) Steganography. (b) Cryptography. (c) Watermarking.

Figure 10: Common procedures of both sides (Alice and Bob) in steganography, cryptography, and watermarking.

(2) Hopper et al. (2002) and Katzenbeisser and Petitcolas (2002) proposed the **complexity-theoretic** definition of steganographic security: A steganographic system is computationally secure if, for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$ in distinguishing between the output of steganography sampler $\mathcal{O}_s$ and the output of random sampler $\mathcal{O}_r$ is negligible.

$$\left|\Pr[\mathcal{A}^{\mathcal{O}_s}(1^\lambda) = 1] - \Pr[\mathcal{A}^{\mathcal{O}_r}(1^\lambda) = 1]\right| < \text{negl}(\lambda), \tag{2}$$

where λ is the security parameter, 1^λ denotes its unary encoding, and $\text{KeyGen}_{\mathcal{D}}(1^\lambda)$ is the key-generation algorithm run on this parameter. The function $\text{negl}(\lambda)$ is negligible in λ, i.e., for every polynomial $p(\cdot)$ there exists an $N \in \mathbb{N}$ such that for all $\lambda > N$ it holds that $\text{negl}(\lambda) < \frac{1}{p(\lambda)}$. Intuitively, this means that any PPT adversary can only distinguish the two oracles with at most negligible advantage as a function of the security parameter.

C Review of Modification-Based Methods

C.1 Review of Format-Based Methods

In this category, the physical features of text symbols are used. Format-based methods do not alter the expression of the original text, so steganographic features are hard to be detected by human-eye perception. These methods can be categorized into space-based, encoding-based, font-based, and spelling-based methods. Note that format-based text steganography does not fall within the scope in the linguistic steganography.

(1) **Space-based** methods use spaces, tabs, or invisible whitespace characters to achieve steganography. Brassil et al. (1995) embed secret messages by shifting the text vertically within a document: if the line spacing is altered, the secret bit "1" is embedded; otherwise, unchanged spacing represents a bit "0". This work also embeds bits by horizontally shifting characters within a line to the left or right. Singh et al. (2012) propose using null spaces in the covertext, where secret bits are hidden by inserting additional null spaces into the plaintext. Por et al. (2012), Mahato et al. (2014), and Kumar et al. (2015) embed data in inter-word, inter-sentence, end-of-line, or inter-paragraph spacing in Microsoft Word documents, achieving high embedding capacity and efficiency with strong visual imperceptibility. Kumar et al. (2016) embed secret bits into whitespace by altering their font type and style. Ramakrishnan et al. (2016) introduce a frequency normalization set combined with character

and string mapping, allowing low-frequency characters to carry multiple hidden bits across different cover characters. Gurunath and Samanta (2023) build a 3-bit covert channel by choosing one of eight concealment operations mapped to symbols 000–111. This method includes inserting one, two, or three spaces between words, and applying small vertical or positional shifts to words or lines.

(2) **Encoding-based** methods use different underlying code points to represent the same visible character. Rahma et al. (2013) embed secret bits in English scripts using Unicode of English alphabet in other languages. Hosmani et al. (2015) propose a dual stage text steganography, making use of Unicode homoglyphs. Naharuddin et al. (2018) embed secret bits in the covertext using the variants of ASCII characters. Ditta et al. (2018) use Arabic text to hide secret messages using a combination of zero-width characters and zero-width joiners of Unicode characters. Taleby Ahvanooey et al. (2018) propose AITSteg, which utilizes encoding variants (such as zero-with characters) to provide end-to-end security during the transmission via social media. Rabevohitra and Li (2019) use a combination and variation of the three primary colors intensities to vehiculate the message when painting non-printing characters. Bashir et al. (2020) use pairs of zero-width characters to represent each 4-bit binary classification that provides a high embedding capacity and reduces computational complexity.

(3) **Font-based** methods use changes in fonts, glyph perturbations, or style as a carrier to hiding information. Khairullah (2009) uses font color of the invisible characters in Microsoft Word documents. Ali et al. (2013) use text file as a carrier for the secret message data where mixed-case fonts are utilized. Ramakrishnan et al. (2016) utilize a font attribute called the character spacing to embed the secret message. Alsaadi et al. (2018) use font color formatting in Microsoft Excel cells as a carrier to hide secret messages. Baawi et al. (2020) aim to improve the method proposed by Rahma et al. (2013) by designing new fonts with special properties to add more factors to hide information.

(4) **Spelling-based** methods hide information by making small spelling variations in words. Shirali-Shahreza (2008) substitutes US and UK spellings to hide secret messages in an English text. Rafat (2009) secretly exchanges real-time or offline information by using words spelled differently.

In summary, despite the simplicity and early popularity of format-based steganographic methods, they suffer from several inherent limitations:

- **Robustness issues.** Operations such as text normalization, document conversion, or copy and paste can easily remove or alter the formatting, leading to irreversible loss of hidden information.
- **Imperceptibility issues.** They are often detectable by automated steganalysis tools, as inconsistencies in spacing, punctuation, or font usage may appear suspicious under scrutiny or statistical analyses (Xiang et al., 2007).
- **Generalizability issues.** Their effectiveness is highly platform-dependent. Formatting information that is preserved in rich text editors (such as Microsoft Word and Microsoft Excel) may be lost in plain text environments. Besides, spelling-based methods are limited to specific languages.

C.2 Review of Lexicon-Based Methods

Lexical modification-based linguistic steganography hides information by modifying individual words or word forms in an existing text, while keeping the semantics and fluency largely intact. Unlike format-based methods that alter surface formatting, lexical methods exploit word-level substitutions or variations. These methods can be categorized into synonym-based methods and masked LM-based methods.

(1) **Synonym-based** methods utilize synonyms as lexical redundancy to embed secret messages. Bolshakov (2005) uses the semantic compatibility of relative synonyms and their collocation words in the context to judge whether the substitution is appropriate to make predictions. Shirali-Shahreza and Shirali-Shahreza (2008) propose a steganographic method for the English text by substituting the words which have different terms in British English and American English. Muhammad et al. (2009) propose a method based on synonym-based substitution, which exploits the simplicity of Bahasa Melayu's grammar rules. Chang and Clark (2010b) embed secret data by contextually substituting synonyms while preserving naturalness and meaning. The method ensures grammatical correctness and contextual appropriateness using n-gram language models, achieving a practical balance between capacity, imperceptibility, and linguistic coherence. Gardiner (2012) achieves the idea of synonym substitution by the WordNet (Miller, 1995)

dictionary in a prototype chat-based steganographic application. Qi et al. (2013) use the Google n-gram corpus and address two main issues: (a) ensuring that a synonym substitution is acceptable in context, and (b) dealing with ambiguity in words with multiple senses so that the encoding of bits remains unambiguous. Chang and Clark (2014) present a practical synonym-substitution stegosystem which takes advantage of WordNet and filters it with a contextual substitution checker. Xiang et al. (2017) propose synonym run-length encoding to prevent the synonym substitution breaking the balance among frequencies of synonyms and improve the statistical undetectability. Xiang et al. (2019) take the word frequency into account, to evade steganalysis detection. Mahato et al. (2020) use familiar synonyms more often to hide more bits and avoid suspicion. Serret et al. (2022) propose a method for messaging applications, and it is based on covertext extension and synonym substitution. Gao et al. (2024a) use an ambiguous token selection algorithm that maintains the textual integrity of the cover messages, and the method is targeted at communication on devices of *Internet of Things*.

(2) **Masked LM-based** methods set some words in an existing text as masked, and utilize masked language models (MLMs) (e.g., BERT (Devlin et al., 2019)) to predict what those tokens might be based on context, and then choose replacements in a way that embeds secret bits. These methods go far beyond the scope of choosing synonyms. Ueoka et al. (2021) first use BERT to replace tokens in the covertext and embed secret messages, and the method is optimized by Zheng and Wu (2022) in an autoregressive way. Zheng et al. (2022) also adopt a similar idea, but with a mechanism of semantic control. Chang (2023) proposes high-probability substitutions at masked positions and encoding message bits by predictive word substitution. Xiang et al. (2023b) propose a causal-perception guided method that constructs a causal perception network by making full use of a trained CNN discriminator to assess the security. Ding et al. (2024b) combine conditional generation-based steganography with substitution-based steganography to achieve a great trade-off between security and capacity. Öztürk et al. (2024) propose a character-level text steganographic method using the MLM feature, embedding bits within a covertext by replacing masked words with BERT predictions in a controlled way.

Lexicon-based linguistic steganographic methods perform well in maintaining the original semantics of the covertext. However, there are several limitations:

- **Capacity issues.** Only words with valid synonyms or lexical variants can be substituted and many are not interchangeable in all situations, so they face contextual constraints.
- **Generalizability issues.** Many of these methods are highly dependent on specific languages, requiring extensive lexical resources such as thesauri or WordNet, which are not available for most other languages.

C.3 Review of Rewriting-Based Methods

Rewriting-based linguistic steganography hides information by altering the syntactic structure of sentences or paraphrasing while preserving meaning, such as reordering constituents, switching from active to passive voice, and inserting subordinate clauses. These methods do not just substitute words (as lexical methods do), but change how words relate grammatically, changing the parse tree or sentence structure. These methods can be categorized into syntax-based methods and paraphrasing-based methods.

(1) **Syntax-based** methods hide information by altering the syntactic structure (i.e. the grammatical arrangement of a sentence) without changing the core meaning. These methods operate at the level of syntax (constituent structure, dependency relations, sentence transformations) rather than simply replacing words or formatting text. Wai and Khine (2011) use a *syntax bank* of possible syntactic structures. They parse cover sentences, look up alternative syntactic forms from the bank, use Shannon-Fano compression, digital signature or syntax transformations, to embed bits via selecting alternate syntax. Barmawi (2016) combines lexical substitution and syntactic transformations to increase payload (embedding capacity) beyond what the lexical method alone achieves. Ou et al. (2025) propose a multi-granularity modification-based linguistic steganography framework that hides information in both syntactic space and symbolic space, enhancing syntactic naturalness and semantic coherence while further increasing embedding capacity.

(2) **Paraphrasing-based** methods use alternative formulations of the same meaning (at the phrase, clause, or sentence level) to hide information. Rather than just swapping synonyms or tweaking formatting, these methods rewrite or transform

parts of the text (via paraphrase rules, templates, or automatically generated paraphrases), choosing among paraphrase options in a way that embeds bits. Kermanidis and Magkos (2009) apply paraphrasing in modern Greek, which first generate paraphrased candidates via several empirical rules, then filter them via supervised learning. The shared key determines which paraphrase should be picked for embedding. Chang and Clark (2010a) use a paraphrase dictionary (Callison-Burch, 2008) and check acceptability using Google n-gram data plus a CCG parser. Paraphrases are then used as transformations to embed secret bits while preserving meaning and fluency. Clarke et al. (2014) hide bits by interactively rewriting sentences until the least-significant bit of a keyed hash chain over the carrier and an "open-stego-key" (or a secret key) matches each target bit, after which the receiver simply replays the same hash chain to deterministically recover the message. Wilson et al. (2014) propose a stegosystem for Twitter, in which a a hierarchical language model is used for unilingual machine translation (paraphrasing) to provide redundancy. This methods is improved by Wilson and Ker (2016) with distortion measures for linguistic steganography. Li et al. (2023a) propose a rewriting-based paradigm, employing BART (Lewis et al., 2020) as the backbone language model to paraphrase the covertext, in which rewriting each token provides redundancy to embed bits. Yang et al. (2024b) propose a sequence-to-sequence-to-sequence (Seq2Seq2Seq) model to achieve the transformation of English-to-German-to-English (En2Ge2En), and embed secret information in the decoding stage of the second Seq2Seq model. This method alters the expression of a given text through pivot translation and semantic-aware bins coding. Xiang et al. (2025) propose a method in which sentence attribute values are used. When they differ from the message to embed, the method automatically generates a paraphrased sentence that matches the target attribute values.

Rewriting-based linguistic steganographic methods are especially strong at preserving the semantics of the covertext and maintaining readability, which in turn often achieves superior imperceptibility compared with other modification-based methods. However, they often face significant drawbacks of **embedding capacity**, particularly when redundancy must come from entire sentences or syntactic structure rather than small lexical spots.

D Reasons of Researching Generative Linguistic Steganography

(1) **High capacity.** Generative language models can utilize the redundancy of the whole vocabulary and flexibly choose a token conditioned on the targeted secret message at each inference position, enabling a much higher embedding rate than modification-based approaches that rely only on limited redundant positions in existing text.

(2) **Advances of language models.** Early generative steganography methods relied on traditional statistical language models such as n-gram algorithms (Muñoz et al., 2010) and hidden Markov models (Moraldo, 2014), which were severely limited in terms of grammatical correctness, fluency and semantic richness. The introduction of neural language models, particularly recurrent neural networks (RNNs) (Yang et al., 2019a), such as long short-term memory (LSTM) (Fang et al., 2017) and gated recurrent unit (GRU) architectures (Yang et al., 2022b), enhanced the coherence and contextual consistency of stegotexts. With the advent of pre-trained generative models built on the transformer architecture (Vaswani et al., 2017), such as GPT-2 (Radford et al., 2019), a decisive breakthrough was achieved over earlier RNN-based models. Unlike sequential architectures that struggled with long-range dependencies and often produced repetitive or incoherent text, transformers leverage self-attention mechanisms to capture global contextual relationships efficiently. This advancement enabled researchers to handle flexible and diverse prompts to generate more natural and human-like stegotexts. A milestone marking the entry into the era of LLMs was the release of OpenAI's GPT-3 (Brown et al., 2020), the first widely recognized model to demonstrate that scaling transformer architectures to hundreds of billions of parameters and training them on massive corpora could achieve unprecedented performance across various tasks. Building on it, subsequent open-weight LLMs are used to generate stegotexts that are both indistinguishable from human writing and adaptable across domains and languages.

(3) **Imperceptibility and security.** The capabilities of LLMs directly support higher imperceptibility. Moreover, recent advances have shown that provably secure steganographic methods (e.g., Zhang et al. (2021b) and Ding et al. (2023)) can be realized within generative frameworks. These approaches equip generative steganography

with strict security guarantees, thereby advancing it beyond empirical validation toward theoretical resilience against steganalysis.

E Review of Modern Generative Linguistic Steganographic Methods

Table 4 presents a detailed taxonomy of modern generative linguistic steganographic methods, which supplements the rough taxonomy shown in Figure 6. It is based on the following six dimensions:

- **Targeted metric(s).** This dimension refers to what kind of improvements are aimed at by the proposed method. Explanations of these three types (security, efficiency, and robustness) are introduced in Section 4.3.
- **Scenario(s).** This dimension refers to the designated scenarios in which the steganographic method is proposed. The methods that can be applied to general generation scenarios are labeled *General*.
- **Backbone model(s).** This dimension refers to each method's language-model requirements or assumptions. Methods that apply across generative autoregressive LMs (for example, those optimizing embedding algorithm and extraction algorithm without relying on a particular architecture) are labeled *Model-agnostic*.
- **Training-free.** This dimension indicates whether the method operates without additional training or fine-tuning of the language model (i.e., using an off-the-shelf LM).
- **Asymmetric.** This dimension indicates whether embedding and extraction are not mirror operations, i.e., the extraction procedure fundamentally differs from embedding. For example, embedding requires LM inference, while extraction does not.
- **Black-box.** This dimension indicates whether the method operates without access to an LM's internal states (e.g., next-token probabilities) and can be implemented through an API-only interaction.

Organized chronologically, Table 4 reveals the following trends in this field:

(1) **Efficiency**, **robustness**, or multi-metric optimization has received growing emphasis.

(2) **Model-agnostic** and **training-free** methods have dominated, driven by LLM compatibility.

(3) **Asymmetric** and **black-box** methods enable lightweight implementations, but almost remain underexplored.

In this section, we review several representative methods for each target metric.

E.1 Toward Higher Security

(1) **Methods based on optimizing modeling.** These methods mark the first wave of neural linguistic steganography emerging alongside neural language models. Fang et al. (2017) are one of the earliest attempts to directly use LSTMs for language modeling to generate stegotexts, and its embedding and extraction are based on block coding. Building on this line, Yang et al. (2019a) formalize the RNN-based embedding paradigm and improve the coding strategy: It ranks next-token probabilities to form a candidate pool, but embeds bits using fixed-length or variable-length Huffman coding. Further, Yang et al. (2021b) discover the *Psic effect*, i.e., perceptual-imperceptibility and statistical-imperceptibility conflict effect in linguistic steganography, and propose a new linguistic steganography based on Variational Auto-Encoder to guarantee both the perceptual-imperceptibility and statistical-imperceptibility.

(2) **Methods toward provable security.** These methods aiming at provable security largely pursue distribution-preserving at each generative step after steganographic encoding. Without considering language modeling, these methods are generally **training-free**. The early baseline is arithmetic coding (AC) (Ziegler et al., 2019), which maps a bitstream to successive token intervals so that sampling closely follows the LM's next-token distribution, yielding low per-step KL distortion and thus strong statistical imperceptibility. Building on this, self-adjusting arithmetic coding (Shen et al., 2020) dynamically trims/reshapes the candidate pool to meet a user-specified imperceptibility (KL divergence) threshold, improving the security–capacity trade-off over vanilla AC. ADG (Adaptive Dynamic Grouping) (Zhang et al., 2021b) aims at provably perfect matching by recursively grouping tokens into equal-probability bins and selecting within bins. For broader cryptographic guarantees under realistic settings, Meteor (Kaptchuk et al., 2021) provides a symmetric-key scheme that achieves computational security while sam-

pling from generative approximations of natural language. iMEC (de Witt et al., 2023) shows that a steganography procedure is perfectly secure under an information-theoretic model of steganography if and only if it is induced by a coupling. To make perfect security practical, Discop (Ding et al., 2023) constructs multiple "distribution copies" and samples from the chosen copy indexed by message bits, preserving the original distribution (**zero KL divergence**) while improving efficiency. Recently, SparSamp (Wang et al., 2025a) further reduces overhead via sparse, message-derived sampling intervals, maintaining provable security but significantly boosting efficiency of embedding and extraction.

(3) **Methods powered by LLMs.** These methods are marked by black-box access, prompt-driven control, or higher-level strategy design beyond the basic token-by-token generation. Wu et al. (2024c) propose LLM-Stega, the first method tailored to **API-only** LLMs: instead of relying on internal logits, it builds keyword sets and an encrypted mapping at the interface level, showing that practical steganography is feasible even without probability access (**black-box**). Lin et al. (2024) propose a **prompt-driven** linguistic steganography with an **in-context learning** framework, explicitly targeting both perceptual and statistical imperceptibility. Most recently, Zhou et al. (2025) frame LLM steganography as a **self-evolving agent system**: an agent loop automatically discovers, composes, and updates a library of embedding strategies at inference time, allowing adaptation to domains and constraints.

E.2 Toward Higher Efficiency

(1) **Methods based on more embedding units.** These methods mainly push linguistic steganography below the word or token level, exploiting characters and script-specific symbols as basic embedding units to boost capacity. Xiang et al. (2020) use an LSTM-based character-level language model to embed at least one bit per character, achieving faster generation and significantly higher embedding rates than earlier word-level neural methods. Adeeb and Kabudian (2022) hide multiple bits per Arabic letter by modulating fine-grained script features such as elongation, pseudo-spaces, and dotted/diacritic patterns, realizing high-capacity, letter-based embedding tailored to Arabic and related scripts. Yan et al. (2024b) apply a tokenization-free character-level scheme with entropy-based candidate construction and lexical emphasis, which substantially reduces out-of-vocabulary cases and improves imperceptibility. Farooq Ahmed Adeeb and Kabudian (2025) combine character-level embedding with foundation LLMs, designing a stegosystem that operates at the character granularity and reports large gains in bits-per-word and bits-per-character over prior neural methods while maintaining comparable security and imperceptibility.

(2) **Methods based on improving entropy or utilization.** These methods mainly aim to make full use of the entropy provided by generative models rather than creating more embedded positions. Pan et al. (2025) revisit prefix-based schemes (e.g., AC/Meteor) and propose two enhancements: security-enhanced and efficiency-enhanced variants. The first optimizes quantization distortion to minimize KL divergence. The second redesigns the sampling mechanism via distribution coupling to improve the steganographic capacity. Bai et al. (2025a) introduce a mechanism of entropy collection that gathers residual entropy left unused in each embedding step and recycles it in subsequent steps, resulting in higher capacity than prior distribution-preserving methods. Pang (2025b) improves entropy utilization in **black-box** settings: instead of requiring access to model's logits, it embeds messages across multiple candidate samples while keeping the generator's output behavior statistically unchanged, achieving provable security and competitive capacity with only API-level access. Wang et al. (2025a) propose sparse, message-driven sampling, which combines the message with pseudo-random numbers to form sampling indices and increasing sampling intervals. It preserves the original probability distribution and introduces only $O(1)$ additional complexity per sampling step and nearly full entropy utilization, enabling the fastest embedding speed without compromising generation speed.

E.3 Toward Higher Robustness

(1) **Methods to tackle tokenization inconsistency.** These methods address incorrect results or failures extraction when the sent tokens are different from received tokens after the **detokenization-retokenization** pipeline between Alice and Bob. Nozaki and Murawaki (2022) first show that subword-based generators can produce multiple valid segments for the same surface text, caus-

ing occasional extraction errors (called **segmentation ambiguity**). They propose simple, language-agnostic disambiguation tricks that prune prefix-confusable tokens so that detokenization is unique without heavily modifying existing systems. Yan et al. (2023) design a distribution-aware pruning strategy that resolves segmentation ambiguity while preserving the probability mass of remaining candidates, improving both security and imperceptibility. The method is realized by finding out a maximum weight independent set (MWIS) in each candidate graph. Yan et al. (2024a) further reduce distortion by adding a verification stage: instead of actively modifying candidate pools, an all-case extraction method is proposed to obtain possible true extracted results. Further, verification is adopted to filter wrong extracted results caused by segmentation ambiguity. Qi et al. (2025a) (SyncPool) then show how to eliminate segmentation ambiguity without sacrificing provable security by grouping all tokens with prefix relationships into ambiguity pools and using a shared CSPRNG to synchronously sample from these pools, keeping both candidate set size and token distribution exactly unchanged. Recently, Yan and Murawaki (2025) generalize from segmentation ambiguity to full tokenization inconsistency between Alice and Bob's models.

(2) **Methods against external modification.** These methods target robustness when a stegotext is perturbed by active tampering. Pang et al. (2025) propose an adaptive robust enhancement framework using a sliding window approach, that can be applied on top of existing generative linguistic steganography schemes. Bai et al. (2025b) present an **asymmetric** framework in which only the encoder has access to the model. Specifically, the encoder uses various permutations of distribution to hide secret bits, while the decoder relies on a sampling function to extract the hidden bits by guessing the permutation used. Guo et al. (2025) integrate classical error-correcting codes into a generative linguistic stegosystem. Qi et al. (2025b) leverage **diffusion language models** to achieve robust and provably security. Their method identifies robust positions in the partially parallel denoising trajectory. Furthermore, they introduce an error correction strategies, including pseudo-random error correction and neighborhood search correction, during steganographic extraction.

F Review of Early Generative Linguistic Steganographic Methods

It is undeniable that early attempts at generative steganography (especially the methods based on n-gram Markov chain before Fang et al. (2017)) provided the initial insights and foundational ideas that inspired many modern methods.

At the first step toward generative linguistic steganography, Wayner (1992) propose the Mimic Functions. The method implements linguistic steganography by generating text that statistically imitates a target language model while embedding hidden data. The method constructs a finite-state grammar or a Markov model that captures the probability distribution of characters or words.

Chapman and Davida (1997) present a system for protecting the privacy of cryptograms to avoid detection by censors. It takes encrypted data (ciphertext) and uses a probabilistic context-free grammar to generate text whose statistical and syntactic patterns resemble ordinary English. The system maps bits from the ciphertext to grammatical production choices, ensuring that each generated sentence is syntactically valid and stylistically natural while deterministically embedding the hidden data. Chapman et al. (2001) propose a fully automated pipeline that turns ciphertext into natural-looking English by combining extensible contextual templates (to fix sentence/paragraph structures) with context-aware synonym replacement driven by large, part-of-speech–classified dictionaries. Chapman and Davida2 (2002) extend the linguistic-steganography system into a deniable-messaging protocol. This method couples automated text generation with layered keys or salts to provide plausible deniability, so that observers cannot prove that any secret beyond the disclosed one exists.

Grothoff et al. (2005) evaluate the potential of covert message transfer in natural language translation that uses automatic machine translation (MT). The key idea is to hide information in the noise that occurs invariably in natural language translation. The method is refined by Stutsman et al. (2006) which requires only the translated text for recovering the hidden bits.

Bergmair and Katzenbeisser (2007) introduce content-aware steganography as a new paradigm, which embeds secrets in the semantic interpretation that a human assigns to a datagram.

Yu et al. (2009) hide data within automatically generated Song-style Ci poetry, leveraging its strict

tonal and rhythmic patterns. The approach maps secret bits to word or phrase selections that satisfy both semantic coherence and the formal prosodic rules of classical Chinese poetry. By combining a rule-based template system with statistical or semantic constraints, the method ensures each generated poem adheres to authentic Ci structures while carrying a high volume of hidden information.

Dai et al. (2010) present a text steganography system for spelling languages based on Markov Chain source model and DES algorithms. By adopting the technique of heuristic composition, this system can produce stegotexts close to nature texts.

Chang and Clark (2010a) is a paraphrasing-based method (discussed in Section C.3), but the nature of utilizing a n-gram model and embedding bits meanwhile generating makes it also belong to generative linguistic steganography. Chang and Clark (2012) propose a linguistic steganography method using word ordering as the linguistic transformation and develop a method using a maximum entropy classifier to determine the naturalness of sentence permutations.

Satir and Isik (2012) employ the lossless LZW data compression algorithm to achieve generative linguistic steganography with high capacity and high security.

Dai et al. (2009) model natural language as a Markov chain and hides data by driving text generation with the chain's state transition probabilities, in which a payload is parsed into bits and at each step. Moraldo (2014) presents a generative steganography scheme that trains a word-level Markov language model on a corpus, then encodes data by iteratively partitioning a numeric range according to each state's outgoing transition probabilities and choosing the next word whose sub-range contains the current data value. Shniperov and Nikitina (2016) extend the idea of utilizing a word-level Markov language model, but the method differs from the existing ones in the more accurate consideration of transition probabilities from one word to another, the ability to use different languages, as well as the use of higher-order Markov chains. Luo et al. (2016) achieves linguistic steganography in the genera of a classic Chinese poem style based on a Markov chain model.

G Review of Linguistic Steganalysis Methods

In this section, we review representative linguistic steganalysis methods and introduce the features of various categories.

G.1 Methods Based on Self-Trained Word Representations

This category mainly includes early methods based on neural modeling methods to extract and analyze text features, consisting of recurrent neural networks (RNNs), convolutional neural networks (CNNs), graph neural networks (GNNs), and hybrid modeling methods. RNN-based detectors typically model the conditional distribution of the next token given previous context, and then use likelihood-based or perplexity-based criteria to identify abnormal sequences introduced by embedding operations. For example, LSTM (Niu et al., 2019; Yang et al., 2019c, 2020a) or GRU (Xu et al., 2023) architectures can be trained on large collections of clean carrier texts to learn typical syntactic and semantic patterns. CNN-based models (Yang et al., 2020c; Xue et al., 2022a; Xu et al., 2023), on the other hand, focus more on local text fragments and are effective at capturing stego-induced changes in short n-grams, such as unusual phrase constructions or statistically rare word combinations. With the introduction of GNNs into linguistic steganalysis, researchers began to represent texts as graphs, where nodes may correspond to words, sentences, or documents, and edges encode semantic similarity, syntactic dependencies, or user–post relationships in social media platforms (Wu et al., 2021; Yang et al., 2023a; Pang et al., 2024b). GNN-based steganalysis then performs message passing over these structures to aggregate multi-hop contextual information, enabling the detection of more global, structure-level anomalies that are difficult to capture using purely sequential models.

G.2 Methods Based on Pre-Trained Word Representations

This category mainly includes modern methods based on pre-trained language models, consisting of masked LMs and generative LMs (or LLMs). Method based on masked language modeling (Peng et al., 2021; Zou et al., 2021; Yang et al., 2022a; Xiang et al., 2022a; Wang et al., 2023a), such as those built on BERT-like architectures, typically exploit the contextualized token representations and token-

level prediction probabilities learned from large-scale unlabeled corpora. In a common pipeline, the pre-trained model is either frozen as a feature extractor or lightly fine-tuned on clean carrier texts, and then a simple classifier (e.g., MLP or shallow CNN) is trained on top of its hidden states to discriminate between stegotexts and covertexts. Generative LMs (or LLMs) extend this idea by modeling the full left-to-right generation process. Instead of relying on hand-crafted or shallow neural features, Yang et al. (2024a) treat linguistic steganalysis as an LLM-based generative task. They fine-tune open LLMs with LoRA for stego vs. cover detection and analyze text distributions via length, perplexity and log-probability. Bai et al. (2024a) treat ChatGPT as a black-box few-shot classifier: supply a prompt with up to 32 labeled examples and ask it whether a new sentence is steganographic or not. Tang et al. (2025) target strongly concealed stegotexts, especially those generated by LLM-based steganography, and proposes an LLM-based steganalysis framework with two operation modes: generation and classification. Wang et al. (2025d) move from isolated sentences to social media settings, and propose STLC-KG, which fuses an LLM with a commonsense knowledge graph to catch logical inconsistencies introduced by steganography.

H Basics and Notions of Steganalysis

According to Simmons' *Prisoners' Problem* (Simmons, 1984) discussed in Section 2.1, in the steganographic scenarios, Eve (the steganalyzer) uses steganalysis methods to distinguish stegotexts from covertexts to determine whether or not she needs to block Alice and Bob's communication. Linguistic steganalysis is a text binary classifier that maps an observed text to a decision on whether it is steganographic or not.

Formally, let $\mathcal{V}$ denote the vocabulary and $\mathcal{X} = \mathcal{V}^*$ the set of all finite-length texts, i.e., $x_i \in \mathcal{X}$. Let $\mathcal{Y} = \{0, 1\}$ be the label set, i.e., $y_i \in \mathcal{Y}$ where $y_i = 0$ indicates that x_i is a covertext and $y_i = 1$ indicates that x_i is a stegotext. The linguistic steganalysis is then modeled as a binary classifier:

$$g(\theta) : \mathcal{X} \to \mathcal{Y},\ g(x;\theta) = \mathbf{1}[f(x;\theta) \geq \tau], \quad (3)$$

in which $f(x;\theta)$ is a scoring function, τ is a decision threshold, and θ is the learnable parameters. Figure 7 shows a sketch of steganalysis. Note that Eve does not need to extract the embedded message and is unable to edit the sent text. Among the classical typology of steganalysis attacks (e.g., stego-only, known-cover, known-stego, and chosen-stego attacks), the online decision problem faced by Eve in our setting is stego-only: at test time she only observes a single text $x \in \mathcal{X}$ and must decide whether $y = 0$ or $y = 1$ without access to the corresponding cover or the embedded message.

I Review of Evaluation Metrics

Following the terminology in Section 4.1, we denote the generated stegotext by the token sequence $[s^{(0)}, \ldots, s^{(T-1)}]$.

I.1 Security

I.1.1 Metrics Based on KL Divergence

According to the information-theoretic perspective of steganography security in Section 2.2, the KL divergence between covertexts and stegotexts matters (Yang et al., 2021b; Zhang et al., 2021b). One typical way to calculate this **text-level** KL divergence is based on a third-party sentence vectorization tool (Le and Mikolov, 2014) to map all covertexts and stegotexts to fixed length dense vectors $\boldsymbol{v}_c$ and $\boldsymbol{v}_s$. Assuming that these vectors obey isotropic Gaussian distribution, the divergence is denoted as:

$$\begin{aligned} D_{\text{KL-text}} &= D_{\text{KL}}(p(\boldsymbol{v}_c) \,\|\, p(\boldsymbol{v}_s)) \\ &\approx \sum (\log \frac{\boldsymbol{\sigma}_s}{\boldsymbol{\sigma}_c} + \frac{\boldsymbol{\sigma}_c^2 + (\boldsymbol{\mu}_c - \boldsymbol{\mu}_s)^2}{2\boldsymbol{\sigma}_s^2} - \frac{1}{2}), \quad (4) \end{aligned}$$

in which $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the mean and standard deviation of sentence vectors. The lower text-level KL divergence indicates greater security.

In addition to the above text-level KL divergence, a formal statistical notion of near-imperceptibility is proposed based on **step-level** KL divergence (Dai and Cai, 2019). Specifically, the average step-level KL divergence is denoted as:

$$D_{\text{KL-step}} = \frac{1}{T} \sum_{t=0}^{T-1} D_{\text{KL}}(\boldsymbol{p}^{(t)} || \hat{\boldsymbol{p}}^{(t)}), \quad (5)$$

in which $\boldsymbol{p}^{(t)}$ is the original probability distribution at t^{th} step, and $\hat{\boldsymbol{p}}^{(t)}$ is transformed from $\boldsymbol{p}^{(t)}$ via steganographic sampling and encoding. The lower step-level KL divergence indicates a higher statistical imperceptibility. Its calculation has two variants: (1) inference based on random-sampling sequences (e.g., theoretical analyses in Dai and Cai (2019)), and (2) inference based on steganographic

sequences (that including its reverse versions are commonly applied) (Ziegler et al., 2019).

Besides, several studies also use KL-derived divergences, most notably the Jensen–Shannon divergence (JSD), to quantify the gap between stegotext and covertext (Yang et al., 2021b; Lin et al., 2024).

I.1.2 Metrics Based on Text Quality

First, one of the most common metrics to evaluate text quality and fluency is perplexity (PPL) (Jelinek et al., 1977). The calculation of PPL is:

$$\begin{aligned}&\mathrm{PPL}([s^{(0)},\dots,s^{(T-1)}])\\&=\exp\left\{-\frac{1}{T-1}\sum_{t=1}^{T-1}\log p_{s^{(t)}}^{(t)}\right\},\end{aligned}\tag{6}$$

in which $\log p_{s^{(t)}}^{(t)}$ is the log-likelihood of the t-th token conditioned on the preceding tokens. The calculated result depends on the adopted evaluation model, and a lower score is better.

Besides, text quality can be measured via similarity between the stegotext (candidate) and its covertext (reference) using BLEU (Papineni et al., 2002; Lin and Och, 2004), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005). BLEU computes the geometric mean of modified n-gram precisions to estimate how similar a candidate text is to one or more reference texts at the lexical n-gram level. ROUGE measures content overlap by counting matching n-grams, skip-bigrams, or longest common subsequences between a candidate and reference to assess how much of the reference's content the candidate preserves. METEOR aligns candidate and reference at the unigram level, then computes a weighted mean of precision and recall with a fragmentation penalty, aiming to better capture semantic adequacy and correlate with human judgments than n-gram precision alone. For these metrics, higher scores indicate greater similarity.

In addition, to evaluate the diversity of stegotexts, the metric called *distinct-n* computes $\frac{|\text{unique } n\text{-grams}|}{|\text{all } n\text{-grams}|}$ over the generated text to quantify lexical diversity (Li et al., 2016). Higher values indicate less repetition and more varied output.

Beyond traditional automatic metrics, human evaluation and LLM-as-a-judge (Gu et al., 2025) are used to rate various dimensions, such as text fluency, naturalness, and logic, which provide human or near-human perceptual assessments.

Note that several studies not only use absolute values to denote text quality, but also use the metric differences between stegotexts and covertexts to denote imperceptibility. For example, the perplexity difference ΔPPL is calculated when the perceptual-imperceptibility and statistical-imperceptibility conflict effect (*Psic Effect*) is taken into account (Yang et al., 2021b).

I.1.3 Metrics Based on Steganalysis

Following the concepts and introduction in Section 5, steganalysis equips Eve with a binary classifier to distinguish stegotext from covertext. The Lower classifier Accuracy, Precision, Recall, or F1 implies a higher imperceptibility of the stegosystem. Let TP, TN, FP, FN be true/false positives/negatives. Accuracy $=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}$, Precision $=\frac{\text{TP}}{\text{TP}+\text{FP}}$, Recall $=\frac{\text{TP}}{\text{TP}+\text{FN}}$, and F1 $=\frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}$. These four indicators are suitable for scenarios in which the positive and negative samples are (approximately) balanced.

I.2 Efficiency

I.2.1 Embedding Capacity

Bits per token (BPT) is the standard measure of embedding capacity (**bit-rate efficiency**), defined as the number of embedded bits divided by the number of tokens in the stegotext: BPT $=\frac{|\mathbf{m}|}{T}$, where $|\mathbf{m}|$ is the bit number of the embedded message $\mathbf{m}$ and T is the token number of the generated stegotext. Some studies report variants, e.g., bits per word (Fang et al., 2017; Yang et al., 2019a), to express capacity at a different granularity.

Besides, entropy utilization is assessed, which normalizes the payload by the generator's information budget and thus enables fair cross-model comparisons. It is defined as

$$\begin{aligned}&\text{Entropy utilization}=\frac{|\mathbf{m}|}{\sum_{t=0}^{T-1}\mathrm{Entropy}(\boldsymbol{p}^{(t)})}\\&=\frac{|\mathbf{m}|}{\sum_{t=0}^{T-1}\sum_{i=1}^{|\mathcal{V}|}-p_i^{(t)}\log_2 p_i^{(t)}},\end{aligned}\tag{7}$$

in which $\boldsymbol{p}^{(t)}$ is the (possibly truncated) distribution used to sample the t-th token, and $|\mathcal{V}|$ is the vocabulary size.

Additionally, the embedding rate ER is used to indicate the percentage of payload bits relative to the total number of bits in the generated stegotext (e.g., under UTF-8 encoding):

$$\mathrm{ER}=\frac{|\mathbf{m}|}{\mathrm{bits}([s^{(0)},\dots,s^{(T-1)}])}\times 100\%.\tag{8}$$

I.2.2 Operation Speed

To assess computational complexity and **runtime efficiency**, some studies report operation-speed metrics, particularly (i) the average time to embed/extract one bit (e.g., ms/bit) and (ii) the average time to embed/extract an entire stegotext (e.g., ms/message).

I.3 Robustness

As there are possible tokenization inconsistencies between Alice and Bob, active attacks, and even computational indeterminism (discussed in Section 7), it is significant to assess whether the extracted message $\mathbf{m}'$ is equal to the embedded message $\mathbf{m}$. The standard metric is the message error rate (MER), which is the ratio of $\mathbf{m}' \neq \mathbf{m}$ cases to all cases: MER $= \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{\mathbf{m}' \neq \mathbf{m}\}$, in which N is the number of stegotext samples. Several variants, such as the bit error rate (BER), are also reported, but length mismatches between $\mathbf{m}'$ and $\mathbf{m}$ can arise from the chosen encoding scheme or from various types of active attack, and length mismatches should be handled explicitly.

J Introduction of Challenges and Future Directions

(1) **Computational indeterminism.** Atil et al. (2025) report that LLMs in hosted environments exhibit computational indeterminism, i.e., the probability distribution over next-token predictions is not fixed even for the same prompt and the same model. When applied to linguistic steganography, this phenomenon substantially undermines the robustness (i.e., extraction success rate) of API-only or black-box steganographic methods.

(2) **Limited embedding capacity.** Compared to steganography based on images, audio, or video, linguistic steganography suffers from an inherently lower embedding capacity, since the smallest manipulable unit is typically a token, word, or character, carrying much more semantic load than a single pixel. Beyond improving entropy utilization, several directions could offer breakthroughs in capacity: (a) integrating steganography with **information retrieval** (Wang et al., 2024a), (b) enhancing **diversity** (Liao et al., 2024), and (c) using **diffusion language models** (Wu et al., 2025; Qi et al., 2025b).

(3) **Tokenization inconsistency.** Even though tokenization inconsistency between Alice and Bob can be prevented from affecting extraction (see Section E.3), existing countermeasures cannot simultaneously achieve provable security and high capacity. In particular, for the provably secure disambiguation method SyncPool (Qi et al., 2025a), the achievable capacity drops to zero when it is required to operate over the entire vocabulary.

(4) **Abrupt termination.** In current experiments, generation usually stops abruptly once the hidden message has been fully embedded (e.g., the samples reported in Ziegler et al. (2019)), producing obviously unnatural stegotext, and the decoded messages often include extra bits. A length-prefixed framing scheme would address this.

(5) **High dependence on white-box LLMs.** According to Table 4, most generative steganographic methods are symmetric, requiring both Alice and Bob to access the same white-box language model for embedding and extraction. As LLMs continue to scale up, their computational and memory demands also grow, pushing such symmetric white-box assumptions further away from realistic deployment scenarios. This suggests that **black-box** and **asymmetric** designs (where only the sender has access to a powerful model, or where interaction is limited to API calls) are particularly promising directions for improving computational efficiency and practicality.

(6) **Challenges of synchronizing prompts.** Fixing a single prompt as a hyperparameter makes steganography inflexible and poorly adapted to different topics, styles, or contexts. On the other hand, synchronizing prompts dynamically (e.g., by sharing or updating them each time) implicitly opens an additional side channel outside the stegotext itself, and this auxiliary channel must also be secured and authenticated. Designing schemes that allow prompt adaptation while avoiding new vulnerabilities therefore remains a non-trivial challenge.

(7) **Unrealistic security evaluations.** Current provably secure methods (Ding et al., 2023; Wang et al., 2025a) focus on minimizing the divergence between stegotexts and LM-generated covertexts, but such distributional guarantees alone do not ensure imperceptibility in realistic, human-facing scenarios. LM-based evaluations (e.g., KL divergence and perplexity) should be regarded only as approximations, which is only justified by the lack of access to the ground-truth distribution. Moreover, as summarized in Figure 11, **human evaluation** and **LLM-as-a-judge** metrics are increasingly crucial in the LLM era, yet they are still missing from the majority of existing works.

(8) **Lack of unity in benchmarks.** For current works of linguistic steganography, there are remarkable inconsistencies in selecting (a) datasets or prompts, (b) language models, (c) steganalysis methods, (d) other metrics.

(9) **Ethics and use restrictions of LLM licenses.** Linguistic steganography faces an unlikely existential threat: LLM licenses. A family of use-restricted licenses, including OpenRAIL and Meta's Llama licenses, explicitly prohibit certain forms of deception, such as representing model use or model outputs as human-generated, which lies at the core of linguistic steganography. The critical perspectives on these licenses come largely from outside the steganography literature. For example, Cui and Araujo (2024) argue that use-restricted licenses sit uneasily with established open-weight principles and are unlikely to provide effective governance on their own. In parallel, work on model openness and on an explicit Open Source AI Definition treats Llama-style and OpenRAIL-style licenses as source-available or open-weight rather than genuinely open source, precisely because they impose behavioral or field-of-use restrictions (White et al., 2025; Open Source Initiative, 2024). We believe that the steganography research community also needs to add its voice to this debate.

Year	Reference	Targeted metric(s)	Scenario(s)	Backbone model(s)	Training-free?	Asymmetric?	Black-box?
2017	Fang et al. (2017)	Security	General	LSTM	✗	✓	✗
2017	Luo and Huang (2017)	Efficiency	Lyric	LSTM	✗	✗	✗
2018	Yang et al. (2018)	Security	General	Markov	✗	✗	✗
2019	Yang et al. (2019a)	Security	General	LSTM	✗	✗	✗
2019	Ziegler et al. (2019)	Security	General	Model-agnostic	✓	✗	✗
2019	Dai and Cai (2019)	Security	General	Model-agnostic	✓	✗	✗
2019	Yang and Ling (2019)	Security	General	Model-agnostic	✓	✗	✗
2019	Tong et al. (2019)	Efficiency	Lyric	LSTM	✗	✗	✗
2020	Yang et al. (2020b)	Security	General	LSTM	✗	✗	✗
2020	Yang et al. (2020d)	Security	General	GRU	✗	✗	✗
2020	Shen et al. (2020)	Security	General	Model-agnostic	✓	✗	✗
2020	Kang et al. (2020)	Security	General	LSTM	✗	✗	✗
2020	Wu et al. (2020)	Security	General	Markov	✗	✗	✗
2020	Xiang et al. (2020)	Efficiency	General	LSTM	✗	✗	✗
2021	Yang et al. (2021b)	Security	General	LSTM, BERT	✗	✗	✗
2021	Zhang et al. (2021b)	Security	General	Model-agnostic	✓	✗	✗
2021	Yang et al. (2021c)	Security	General	LSTM, GRU, Transformer	✗	✗	✗
2021	Kaptchuk et al. (2021)	Security	General	Model-agnostic	✓	✗	✗
2021	Zhang et al. (2021a)	Security	General	Transformer	✗	✗	✗
2021	Li et al. (2021)	Security	General	LSTM, Transformer	✗	✗	✗
2021	Yang et al. (2021a)	Security	General	LSTM	✗	✗	✗
2021	Wang et al. (2021)	Security	Lyric	LSTM	✗	✗	✗
2022	Yang et al. (2022b)	Security	Lyric	GRU	✗	✗	✗
2022	Yu et al. (2022)	Security	General	Model-agnostic	✓	✗	✗
2022	Zhou et al. (2022)	Security	General	LSTM, CNN	✗	✗	✗
2022	Yi et al. (2022a)	Security/Efficiency	General	BERT	✓	✗	✗
2022	Varol Arısoy (2022)	Efficiency	General	LSTM	✗	✗	✗
2022	Adeeb and Kabudian (2022)	Efficiency	Arabic text	LSTM	✗	✗	✗
2022	Nozaki and Murawaki (2022)	Robustness	General	Model-agnostic	✓	✗	✗
2023	Sun et al. (2023)	Security	General	T5	✗	✗	✗
2023	Lu et al. (2023a)	Security	General	Model-agnostic	✓	✗	✗
2023	Wang et al. (2023b)	Security	General	Transformer	✗	✗	✗
2023	Ding et al. (2023)	Security	General	Model-agnostic	✓	✗	✗
2023	Wang et al. (2023c)	Security	General	BART, GPT-2	✓	✗	✗
2023	Wang et al. (2023d)	Security	Image-to-text	ClipCap	✗	✗	✗
2023	de Witt et al. (2023)	Security	General	Model-agnostic	✓	✗	✗
2023	Yan et al. (2023)	Security/Robustness	General	Model-agnostic	✓	✗	✗
2023	Lu et al. (2023b)	Security/Robustness	Translation	Transformer	✗	✗	✗
2023	Peng et al. (2023b)	Robustness	Image-to-text	ResNet-101, LSTM	✗	✗	✗
2024	Zhang et al. (2024)	Security	Summarization	BERT, Conditional Random Field (CRF)	✓	✗	✗
2024	Wu et al. (2024c)	Security	General	Model-agnostic	✓	✓	✓
2024	Xiang et al. (2024)	Security	Translation	BERT, LSTM	✗	✗	✗
2024	Lin et al. (2024)	Security	General	Model-agnostic	✓	✗	✗
2024	Li et al. (2024b)	Security	General	Model-agnostic	✓	✗	✗
2024	Liu et al. (2024)	Security	Live comments	ResNet, LSTM	✗	✗	✗
2024	Namitha et al. (2024)	Security	General	Model-agnostic	✓	✗	✗
2024	Pang et al. (2024a)	Security	General	Model-agnostic	✓	✗	✗
2024	Li et al. (2024a)	Security	Group chat	BERT, Transformer	✓	✗	✗
2024	Cao et al. (2024)	Security	General	Model-agnostic	✓	✗	✗
2024	Chen et al. (2024)	Security	General	Model-agnostic	✗	✗	✗
2024	Qin et al. (2024)	Security	General	Model-agnostic	✓	✗	✗
2024	Lin and Wang (2024)	Security	Translation	LSTM	✗	✗	✗
2024	Yan et al. (2024b)	Security/Efficiency	General	ByGPT5	✓	✗	✗
2024	Steinebach (2024)	Efficiency	General	Model-agnostic	✓	✓	✓
2024	Wang et al. (2024a)	Efficiency	Social media	Model-agnostic	✓	✗	✗
2024	Liao et al. (2024)	Efficiency	Social media	Model-agnostic	✓	✗	✗
2024	Huang et al. (2024)	Security/Robustness	General	Model-agnostic	✓	✗	✗
2024	Yan et al. (2024a)	Security/Robustness	General	Model-agnostic	✓	✗	✗
2024	Bai et al. (2024b)	Efficiency/Robustnes	General	Model-agnostic	✓	✗	✓
2025	Wang et al. (2025b)	Security	General	Model-agnostic	✓	✗	✗
2025	Liao et al. (2025)	Security	General	Model-agnostic	✓	✗	✗
2025	Woźniak et al. (2025)	Security	General	Model-agnostic	✓	✗	✗
2025	Long et al. (2025)	Security	General	Model-agnostic	✓	✗	✗
2025	Zhang et al. (2025a)	Security	General	Model-agnostic	✓	✗	✗
2025	Chen et al. (2025a)	Security	General	Model-agnostic	✓	✗	✗
2025	Shi et al. (2025)	Security	General	XLM-RoBERTa, Transformer	✓	✗	✗
2025	Liu et al. (2025)	Security	General	Model-agnostic	✓	✗	✗
2025	Kuznetsov et al. (2025)	Security	General	Model-agnostic	✓	✗	✗
2025	Zhou et al. (2025)	Security	General	Multi-agent	✓	✓	✗
2025	Pan et al. (2025)	Security/Efficiency	General	Model-agnostic	✓	✗	✗
2025	Pang (2025a)	Security/Efficiency	General	Model-agnostic	✓	✗	✗
2025	Wang et al. (2025a)	Security/Efficiency	General	Model-agnostic	✓	✗	✗
2025	Bai et al. (2025a)	Security/Efficiency	General	Model-agnostic	✓	✗	✗
2025	Pang (2025b)	Efficiency	General	Model-agnostic	✓	✗	✓
2025	Huang et al. (2025b)	Efficiency	General	Model-agnostic	✓	✗	✗
2025	Wu et al. (2025)	Efficiency	General	Diffusion language model	✗	✗	✗
2025	Li et al. (2025b)	Efficiency	General	Multi-agent	✗	✗	✓
2025	Farooq Ahmed Adeeb and Kabudian (2025)	Efficiency	General	Model-agnostic	✓	✗	✗
2025	Yan and Murawaki (2025)	Security/Robustness	General	Model-agnostic	✓	✗	✗
2025	Bai et al. (2025b)	Security/Robustness	General	Model-agnostic	✓	✓	✗
2025	Qi et al. (2025a)	Security/Robustness	General	Model-agnostic	✓	✗	✗
2025	Qi et al. (2025b)	Security/Robustness	General	Diffusion language model	✓	✗	✗
2025	Feng et al. (2025)	Efficiency/Robustnes	General	Model-agnostic	✓	✗	✗
2025	Perry et al. (2025)	Robustness	General	Model-agnostic	✓	✗	✗
2025	Pang et al. (2025)	Robustness	General	Model-agnostic	✓	✗	✗
2025	Guo et al. (2025)	Robustness	General	Model-agnostic	✓	✗	✗
2025	Chen et al. (2025b)	Robustness	General	Model-agnostic	✓	✗	✗

Table 4: Taxonomy for generative linguistic steganographic methods based on methodology features.

Year	Reference	Backbone model	Based on pre-trained model(s)?	Experiments on IMDB dataset (Maas et al., 2011)?	Experiments on Twitter dataset (Go et al., 2009)?	Experiments on News dataset (Kaggle, 2017)?	Experiments on other dataset(s)?
2019	Yang et al. (2019b)	Feed-forward neural networks	✗	✗	✗	✗	✓
2019	Niu et al. (2019)	Bi-LSTMs; CNNs	✗	✓	✓	✗	✗
2019	Yang et al. (2019c)	Bi-RNNs; LSTMs	✗	✓	✓	✗	✗
2019	Wen et al. (2019)	Word2vec; CNNs	✓	✗	✗	✗	✓
2020	Yang et al. (2020c)	CNNs	✗	✗	✗	✗	✓
2020	Yang et al. (2020a)	Bi-LSTMs	✗	✓	✓	✓	✗
2020	Bao et al. (2020)	Bi-LSTMs; CNNs	✗	✓	✓	✓	✗
2020	Li et al. (2020)	Bi-LSTMs; GNNs	✗	✗	✗	✗	✓
2021	Wu et al. (2021)	GNNs	✗	✓	✓	✗	✗
2021	Jiao et al. (2021)	Attention	✗	✓	✓	✓	✗
2021	Peng et al. (2021)	BERT; Bi-LSTMs; CNNs	✓	✓	✓	✗	✗
2021	Zou et al. (2021)	BERT; Bi-LSTMs	✓	✗	✓	✓	✗
2021	Xu et al. (2021)	BERT; Bi-LSTMs	✓	✓	✓	✓	✗
2022	Fu et al. (2022)	GNNs	✗	✓	✓	✗	✗
2022	Yi et al. (2022b)	LSTMs	✗	✓	✓	✗	✗
2022	Xue et al. (2022a)	CNNs	✗	✓	✓	✓	✗
2022	Yang et al. (2022a)	BERT; GNNs	✓	✓	✓	✓	✗
2022	Guo et al. (2022)	BERT	✓	✓	✓	✓	✗
2022	Xiang et al. (2022a)	BERT; GNNs	✓	✓	✓	✗	✗
2022	Xue et al. (2022b)	BERT; CNNs	✓	✓	✓	✓	✗
2022	Xu et al. (2022)	BERT; Bi-GRUs; CNNs	✓	✓	✓	✓	✗
2022	Wen et al. (2022b)	BERT; Bi-LSTMs	✓	✓	✓	✓	✗
2022	Wen et al. (2022a)	BERT	✓	✓	✓	✓	✗
2023	Yang et al. (2023b)	GNNs	✗	✗	✗	✗	✓
2023	Xu et al. (2023)	CNNs; Bi-GRU	✗	✗	✗	✗	✓
2023	Yang et al. (2023a)	GNNs	✗	✓	✓	✗	✗
2023	Li et al. (2023b)	Attention; CNNs	✗	✓	✓	✗	✗
2023	Wen et al. (2023a)	Bi-LSTMs; CNNs	✗	✓	✓	✓	✓
2023	Xiang et al. (2023a)	CNNs	✗	✗	✗	✗	✓
2023	Peng et al. (2023a)	BERT	✓	✓	✓	✗	✗
2023	Wang et al. (2023e)	GloVe; LSTMs	✓	✗	✗	✗	✓
2023	Wang et al. (2023f)	BERT	✓	✓	✓	✗	✗
2023	Wen et al. (2023b)	BERT	✓	✓	✓	✓	✗
2023	Wang et al. (2023a)	BERT	✓	✓	✓	✓	✗
2024	Pang et al. (2024b)	GNNs	✗	✗	✓	✗	✓
2024	Guo et al. (2024)	Model-agnostic	✗	✓	✓	✓	✗
2024	Wang et al. (2024b)	CNNs; Attention	✗	✓	✓	✓	✗
2024	Yang et al. (2024a)	LLM (model-agnostic); LoRA	✓	✓	✓	✓	✗
2024	Bai et al. (2024a)	LLM (model-agnostic)	✓	✓	✓	✗	✗
2024	Xue et al. (2024)	BERT; CNNs	✓	✓	✓	✓	✗
2024	Yuan et al. (2024)	BERT; Bi-LSTMs	✓	✓	✓	✓	✗
2024	Wu et al. (2024b)	BERT	✓	✓	✓	✓	✗
2024	Gao et al. (2024b)	ELECTRA (Gutmann and Hyvärinen, 2010)	✓	✓	✓	✓	✗
2024	You et al. (2024)	BERT	✓	✓	✓	✗	✗
2025	Huang et al. (2025a)	GNNs	✗	✗	✓	✗	✓
2025	Lu et al. (2025)	CNNs; GNNs; Attention	✗	✗	✗	✗	✓
2025	Tang et al. (2025)	LLM (model-agnostic); LoRA	✓	✓	✓	✓	✗
2025	Wang et al. (2025d)	LLM (model-agnostic); GNNs	✓	✓	✓	✓	✗
2025	Xiao et al. (2025)	RoBERTa; CNNs	✓	✓	✓	✓	✗
2025	Zhang et al. (2025b)	BERT; CNNs	✓	✓	✓	✓	✗
2025	Yang et al. (2025)	ERNIE (Sun et al., 2019); Attention	✓	✓	✓	✓	✗
2025	Li et al. (2025a)	Word2vec; LSTMs; CNNs	✓	✓	✓	✓	✗
2025	Zhang et al. (2025c)	BERT	✓	✓	✓	✓	✗
2025	Wang et al. (2025c)	BERT; GNNs	✓	✗	✗	✗	✓
2025	Luo et al. (2025b)	BERT; Bi-LSTMs	✓	✓	✓	✓	✗
2025	Luo et al. (2025a)	BERT; Bi-LSTMs	✓	✓	✓	✓	✗
2025	Niu and Yang (2025)	RoBERTa; Bi-RNNs	✓	✗	✗	✗	✓
2026	Wang et al. (2026)	BERT	✓	✗	✗	✗	✓

Table 5: Methodology features and experiment datasets of linguistic steganalysis methods.

Year	Reference	FCN (Yang et al., 2019b)	CNN (Wen et al., 2019)	R-BiLSTM-C (Niu et al., 2019)	TS-RNN (Yang et al., 2019c)	TS-CSW (Yang et al., 2020c)	Other
2017	Fang et al. (2017)	✗	✗	✗	✗	✗	✗
2017	Luo and Huang (2017)	✗	✗	✗	✗	✗	✗
2018	Yang et al. (2018)	✗	✗	✗	✗	✗	✓
2019	Yang et al. (2019a)	✗	✗	✗	✗	✗	✓
2019	Ziegler et al. (2019)	✗	✗	✗	✗	✗	✗
2019	Dai and Cai (2019)	✗	✗	✗	✗	✗	✗
2019	Yang and Ling (2019)	✗	✗	✗	✗	✗	✓
2019	Tong et al. (2019)	✗	✗	✗	✗	✗	✗
2020	Yang et al. (2020b)	✓	✗	✗	✗	✗	✗
2020	Yang et al. (2020d)	✗	✗	✗	✗	✗	✗
2020	Shen et al. (2020)	✗	✗	✗	✗	✗	✗
2020	Kang et al. (2020)	✗	✗	✗	✗	✗	✓
2020	Wu et al. (2020)	✗	✗	✗	✗	✗	✓
2020	Xiang et al. (2020)	✗	✗	✗	✗	✗	✗
2021	Yang et al. (2021b)	✓	✓	✗	✗	✗	✓
2021	Zhang et al. (2021b)	✓	✗	✗	✗	✓	✗
2021	Yang et al. (2021c)	✓	✗	✗	✓	✓	✗
2021	Kaptchuk et al. (2021)	✗	✗	✗	✗	✗	✗
2021	Zhang et al. (2021a)	✓	✗	✗	✗	✓	✗
2021	Li et al. (2021)	✓	✗	✗	✗	✗	✗
2021	Yang et al. (2021a)	✗	✓	✗	✓	✗	✗
2021	Wang et al. (2021)	✗	✗	✗	✗	✗	✗
2022	Yang et al. (2022b)	✗	✗	✗	✗	✗	✗
2022	Yu et al. (2022)	✗	✗	✗	✗	✗	✗
2022	Zhou et al. (2022)	✗	✓	✓	✓	✗	✗
2022	Yi et al. (2022a)	✗	✗	✗	✓	✗	✓
2022	Varol Arısoy (2022)	✓	✓	✗	✗	✗	✓
2022	Adeeb and Kabudian (2022)	✗	✗	✗	✗	✗	✗
2022	Nozaki and Murawaki (2022)	✗	✗	✗	✗	✗	✓
2023	Sun et al. (2023)	✓	✓	✓	✗	✗	✗
2023	Lu et al. (2023a)	✗	✗	✓	✓	✓	✗
2023	Wang et al. (2023b)	✗	✗	✗	✗	✗	✓
2023	Ding et al. (2023)	✓	✗	✓	✗	✗	✓
2023	Wang et al. (2023c)	✗	✗	✓	✗	✗	✓
2023	Wang et al. (2023d)	✓	✗	✗	✗	✗	✗
2023	de Witt et al. (2023)	✗	✗	✗	✗	✗	✗
2023	Yan et al. (2023)	✗	✗	✗	✗	✗	✓
2023	Lu et al. (2023b)	✗	✗	✓	✓	✓	✗
2023	Peng et al. (2023b)	✗	✓	✓	✗	✗	✓
2024	Zhang et al. (2024)	✗	✗	✗	✗	✓	✓
2024	Wu et al. (2024c)	✗	✓	✗	✗	✗	✓
2024	Xiang et al. (2024)	✗	✓	✓	✓	✗	✗
2024	Lin et al. (2024)	✗	✗	✓	✗	✗	✓
2024	Li et al. (2024b)	✗	✗	✓	✗	✓	✗
2024	Liu et al. (2024)	✗	✗	✗	✗	✗	✗
2024	Namitha et al. (2024)	✗	✗	✗	✗	✗	✓
2024	Pang et al. (2024a)	✗	✗	✗	✗	✓	✗
2024	Li et al. (2024a)	✗	✗	✗	✗	✓	✗
2024	Cao et al. (2024)	✗	✗	✗	✓	✗	✗
2024	Chen et al. (2024)	✗	✗	✗	✗	✓	✗
2024	Qin et al. (2024)	✗	✗	✗	✗	✗	✓
2024	Lin and Wang (2024)	✗	✓	✗	✗	✗	✓
2024	Yan et al. (2024b)	✗	✗	✗	✗	✗	✓
2024	Steinebach (2024)	✗	✗	✗	✗	✗	✗
2024	Wang et al. (2024a)	✗	✗	✗	✗	✗	✓
2024	Liao et al. (2024)	✗	✗	✗	✗	✗	✗
2024	Huang et al. (2024)	✗	✗	✗	✗	✗	✗
2024	Yan et al. (2024a)	✗	✗	✗	✗	✗	✓
2024	Bai et al. (2024b)	✗	✗	✗	✗	✗	✗
2025	Wang et al. (2025b)	✗	✓	✗	✗	✓	✓
2025	Liao et al. (2025)	✗	✗	✗	✗	✓	✗
2025	Woźniak et al. (2025)	✗	✗	✗	✗	✗	✗
2025	Long et al. (2025)	✓	✓	✓	✗	✗	✗
2025	Zhang et al. (2025a)	✓	✗	✓	✗	✗	✓
2025	Chen et al. (2025a)	✓	✓	✓	✓	✗	✓
2025	Shi et al. (2025)	✗	✓	✗	✓	✗	✓
2025	Liu et al. (2025)	✗	✗	✗	✗	✗	✓
2025	Kuznetsov et al. (2025)	✗	✗	✗	✓	✗	✓
2025	Zhou et al. (2025)	✗	✓	✗	✗	✗	✓
2025	Pan et al. (2025)	✓	✗	✓	✗	✗	✓
2025	Pang (2025a)	✗	✗	✓	✓	✓	✓
2025	Wang et al. (2025a)	✓	✗	✓	✗	✗	✓
2025	Bai et al. (2025a)	✗	✗	✗	✗	✗	✗
2025	Pang (2025b)	✗	✗	✗	✗	✓	✗
2025	Huang et al. (2025b)	✗	✗	✗	✗	✗	✗
2025	Wu et al. (2025)	✗	✓	✗	✓	✗	✓
2025	Li et al. (2025b)	✓	✗	✗	✗	✗	✓
2025	Farooq Ahmed Adeeb and Kabudian (2025)	✗	✗	✗	✗	✗	✓
2025	Yan and Murawaki (2025)	✗	✗	✗	✗	✗	✓
2025	Bai et al. (2025b)	✗	✗	✗	✗	✗	✗
2025	Qi et al. (2025a)	✗	✗	✗	✗	✗	✗
2025	Qi et al. (2025b)	✓	✗	✓	✗	✗	✓
2025	Feng et al. (2025)	✗	✗	✗	✗	✗	✗
2025	Perry et al. (2025)	✗	✗	✗	✗	✗	✗
2025	Pang et al. (2025)	✗	✗	✗	✗	✓	✗
2025	Guo et al. (2025)	✗	✗	✗	✗	✗	✗
2025	Chen et al. (2025b)	✓	✗	✗	✓	✗	✓

Table 6: Adopted steganalysis methods to evaluate imperceptibility of the generative linguistic steganographic methods.

Year	Reference	PPL	KLD (step-level)	KLD (text-level)	BLEU	ROUGE	METEOR	Diversity	Human evaluation	LLM-as-a-judge	Other
2017	Fang et al. (2017)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2017	Luo and Huang (2017)	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
2018	Yang et al. (2018)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2019	Yang et al. (2019a)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
2019	Ziegler et al. (2019)	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗
2019	Dai and Cai (2019)	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
2019	Yang and Ling (2019)	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗
2019	Tong et al. (2019)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
2020	Yang et al. (2020b)	✓	✗	✗	✓	✓	✗	✗	✗	✗	✓
2020	Yang et al. (2020d)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2020	Shen et al. (2020)	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗
2020	Kang et al. (2020)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
2020	Wu et al. (2020)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2020	Xiang et al. (2020)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2021	Yang et al. (2021b)	✓	✗	✓	✗	✗	✗	✗	✓	✗	✗
2021	Zhang et al. (2021b)	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗
2021	Yang et al. (2021c)	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗
2021	Kaptchuk et al. (2021)	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
2021	Zhang et al. (2021a)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2021	Li et al. (2021)	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗
2021	Yang et al. (2021a)	✓	✗	✓	✓	✗	✗	✓	✗	✗	✗
2021	Wang et al. (2021)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
2022	Yang et al. (2022b)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
2022	Yu et al. (2022)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2022	Zhou et al. (2022)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
2022	Yi et al. (2022a)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2022	Varol Arısoy (2022)	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
2022	Adeeb and Kabudian (2022)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
2022	Nozaki and Murawaki (2022)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
2023	Sun et al. (2023)	✓	✗	✗	✓	✓	✗	✗	✗	✗	✓
2023	Lu et al. (2023a)	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
2023	Wang et al. (2023b)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
2023	Ding et al. (2023)	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
2023	Wang et al. (2023c)	✓	✗	✓	✓	✓	✗	✗	✗	✗	✓
2023	Wang et al. (2023d)	✓	✗	✗	✓	✗	✓	✓	✗	✗	✗
2023	de Witt et al. (2023)	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
2023	Yan et al. (2023)	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
2023	Lu et al. (2023b)	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
2023	Peng et al. (2023b)	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗
2024	Zhang et al. (2024)	✗	✗	✗	✗	✓	✓	✗	✗	✗	✓
2024	Wu et al. (2024c)	✓	✗	✓	✗	✗	✗	✗	✓	✗	✓
2024	Xiang et al. (2024)	✓	✓	✗	✓	✗	✗	✗	✗	✗	✓
2024	Lin et al. (2024)	✓	✗	✓	✗	✗	✗	✗	✗	✓	✗
2024	Li et al. (2024b)	✓	✗	✗	✗	✓	✗	✗	✗	✗	✓
2024	Liu et al. (2024)	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗
2024	Namitha et al. (2024)	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗
2024	Pang et al. (2024a)	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗
2024	Li et al. (2024a)	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗
2024	Cao et al. (2024)	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗
2024	Chen et al. (2024)	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
2024	Qin et al. (2024)	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗
2024	Lin and Wang (2024)	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗
2024	Yan et al. (2024b)	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓
2024	Steinebach (2024)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
2024	Wang et al. (2024a)	✓	✗	✗	✗	✗	✗	✓	✗	✗	✓
2024	Liao et al. (2024)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
2024	Huang et al. (2024)	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗
2024	Yan et al. (2024a)	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
2024	Bai et al. (2024b)	✓	✗	✗	✗	✗	✗	✓	✗	✓	✗
2025	Wang et al. (2025b)	✓	✗	✓	✓	✗	✗	✗	✗	✗	✓
2025	Liao et al. (2025)	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
2025	Woźniak et al. (2025)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
2025	Long et al. (2025)	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗
2025	Zhang et al. (2025a)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
2025	Chen et al. (2025a)	✓	✗	✓	✗	✗	✗	✓	✗	✗	✓
2025	Shi et al. (2025)	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗
2025	Liu et al. (2025)	✓	✗	✓	✗	✗	✗	✗	✓	✗	✗
2025	Kuznetsov et al. (2025)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
2025	Zhou et al. (2025)	✓	✗	✓	✗	✗	✗	✗	✓	✗	✓
2025	Pan et al. (2025)	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
2025	Pang (2025a)	✓	✗	✗	✗	✗	✗	✓	✗	✗	✓
2025	Wang et al. (2025a)	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
2025	Bai et al. (2025a)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2025	Pang (2025b)	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗
2025	Huang et al. (2025b)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
2025	Wu et al. (2025)	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
2025	Li et al. (2025b)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
2025	Farooq Ahmed Adeeb and Kabudian (2025)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2025	Yan and Murawaki (2025)	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
2025	Bai et al. (2025b)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2025	Qi et al. (2025a)	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
2025	Qi et al. (2025b)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
2025	Feng et al. (2025)	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
2025	Perry et al. (2025)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
2025	Pang et al. (2025)	✓	✗	✗	✗	✓	✗	✓	✗	✗	✗
2025	Guo et al. (2025)	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
2025	Chen et al. (2025b)	✓	✗	✓	✗	✗	✗	✓	✗	✗	✓

Table 7: Adopted metrics (excluding steganalysis) to evaluate steganographic security, including the imperceptibility, and the text quality of stegotexts based on generative linguistic steganographic methods.

Year	Reference	Bits per token	Entropy utilization	Embedding rate	Operation speed	Error rate
2017	Fang et al. (2017)	✓	✗	✗	✗	✗
2017	Luo and Huang (2017)	✗	✗	✓	✗	✗
2018	Yang et al. (2018)	✓	✗	✓	✗	✗
2019	Yang et al. (2019a)	✓	✗	✓	✓	✗
2019	Ziegler et al. (2019)	✓	✗	✗	✗	✗
2019	Dai and Cai (2019)	✗	✗	✗	✗	✗
2019	Yang and Ling (2019)	✓	✗	✗	✗	✗
2019	Tong et al. (2019)	✓	✗	✗	✗	✗
2020	Yang et al. (2020b)	✓	✗	✓	✗	✗
2020	Yang et al. (2020d)	✓	✗	✗	✗	✗
2020	Shen et al. (2020)	✓	✗	✗	✓	✗
2020	Kang et al. (2020)	✓	✗	✗	✗	✗
2020	Wu et al. (2020)	✗	✗	✓	✓	✗
2020	Xiang et al. (2020)	✗	✗	✓	✓	✗
2021	Yang et al. (2021b)	✓	✗	✗	✗	✗
2021	Zhang et al. (2021b)	✓	✗	✗	✗	✗
2021	Yang et al. (2021c)	✗	✗	✗	✗	✗
2021	Kaptchuk et al. (2021)	✓	✗	✗	✓	✗
2021	Zhang et al. (2021a)	✗	✗	✗	✗	✗
2021	Li et al. (2021)	✓	✗	✗	✗	✗
2021	Yang et al. (2021a)	✓	✗	✗	✓	✗
2021	Wang et al. (2021)	✓	✗	✗	✓	✗
2022	Yang et al. (2022b)	✓	✗	✓	✗	✗
2022	Yu et al. (2022)	✓	✗	✗	✓	✗
2022	Zhou et al. (2022)	✓	✗	✓	✗	✗
2022	Yi et al. (2022a)	✗	✗	✗	✗	✗
2022	Varol Arısoy (2022)	✓	✗	✓	✓	✗
2022	Adeeb and Kabudian (2022)	✗	✗	✓	✗	✗
2022	Nozaki and Murawaki (2022)	✓	✗	✗	✗	✓
2023	Sun et al. (2023)	✗	✗	✓	✗	✗
2023	Lu et al. (2023a)	✓	✗	✗	✗	✓
2023	Wang et al. (2023b)	✓	✗	✗	✗	✗
2023	Ding et al. (2023)	✓	✓	✗	✓	✗
2023	Wang et al. (2023c)	✓	✗	✗	✗	✗
2023	Wang et al. (2023d)	✓	✗	✗	✗	✗
2023	de Witt et al. (2023)	✗	✓	✗	✓	✓
2023	Yan et al. (2023)	✓	✗	✗	✓	✓
2023	Lu et al. (2023b)	✗	✗	✗	✗	✓
2023	Peng et al. (2023b)	✓	✗	✗	✗	✓
2024	Zhang et al. (2024)	✗	✗	✗	✓	✗
2024	Wu et al. (2024c)	✓	✗	✗	✗	✗
2024	Xiang et al. (2024)	✓	✗	✗	✗	✗
2024	Lin et al. (2024)	✓	✗	✗	✗	✗
2024	Li et al. (2024b)	✓	✗	✗	✗	✗
2024	Liu et al. (2024)	✓	✗	✗	✗	✗
2024	Namitha et al. (2024)	✓	✗	✗	✓	✓
2024	Pang et al. (2024a)	✗	✗	✗	✗	✗
2024	Li et al. (2024a)	✗	✗	✗	✗	✗
2024	Cao et al. (2024)	✓	✗	✓	✓	✗
2024	Chen et al. (2024)	✓	✗	✗	✗	✗
2024	Qin et al. (2024)	✗	✗	✗	✗	✗
2024	Lin and Wang (2024)	✓	✗	✗	✗	✗
2024	Yan et al. (2024b)	✗	✗	✓	✗	✗
2024	Steinebach (2024)	✓	✗	✗	✗	✓
2024	Wang et al. (2024a)	✓	✗	✗	✗	✗
2024	Liao et al. (2024)	✓	✗	✗	✗	✗
2024	Huang et al. (2024)	✓	✗	✗	✗	✓
2024	Yan et al. (2024a)	✓	✗	✗	✗	✓
2024	Bai et al. (2024b)	✓	✗	✗	✗	✓
2025	Wang et al. (2025b)	✓	✗	✗	✗	✗
2025	Liao et al. (2025)	✓	✓	✗	✓	✗
2025	Woźniak et al. (2025)	✓	✗	✗	✗	✗
2025	Long et al. (2025)	✓	✗	✗	✗	✗
2025	Zhang et al. (2025a)	✗	✗	✗	✗	✗
2025	Chen et al. (2025a)	✓	✗	✗	✗	✗
2025	Shi et al. (2025)	✓	✗	✗	✗	✗
2025	Liu et al. (2025)	✓	✗	✗	✗	✗
2025	Kuznetsov et al. (2025)	✗	✗	✓	✓	✓
2025	Zhou et al. (2025)	✓	✗	✗	✗	✗
2025	Pan et al. (2025)	✓	✓	✗	✓	✗
2025	Pang (2025a)	✓	✗	✗	✓	✗
2025	Wang et al. (2025a)	✓	✓	✗	✓	✓
2025	Bai et al. (2025a)	✓	✓	✗	✓	✗
2025	Pang (2025b)	✓	✓	✗	✓	✗
2025	Huang et al. (2025b)	✓	✗	✗	✗	✗
2025	Wu et al. (2025)	✓	✗	✗	✓	✓
2025	Li et al. (2025b)	✗	✗	✗	✗	✓
2025	Farooq Ahmed Adeeb and Kabudian (2025)	✓	✗	✗	✗	✗
2025	Yan and Murawaki (2025)	✓	✗	✗	✓	✓
2025	Bai et al. (2025b)	✓	✗	✗	✓	✓
2025	Qi et al. (2025a)	✓	✓	✗	✓	✓
2025	Qi et al. (2025b)	✓	✗	✗	✓	✓
2025	Feng et al. (2025)	✓	✗	✗	✗	✓
2025	Perry et al. (2025)	✗	✗	✗	✗	✓
2025	Pang et al. (2025)	✓	✗	✗	✗	✓
2025	Guo et al. (2025)	✗	✗	✗	✗	✓
2025	Chen et al. (2025b)	✓	✗	✗	✗	✓

Table 8: Adopted metrics to evaluate efficiency and robustness of the generative linguistic steganographic methods.

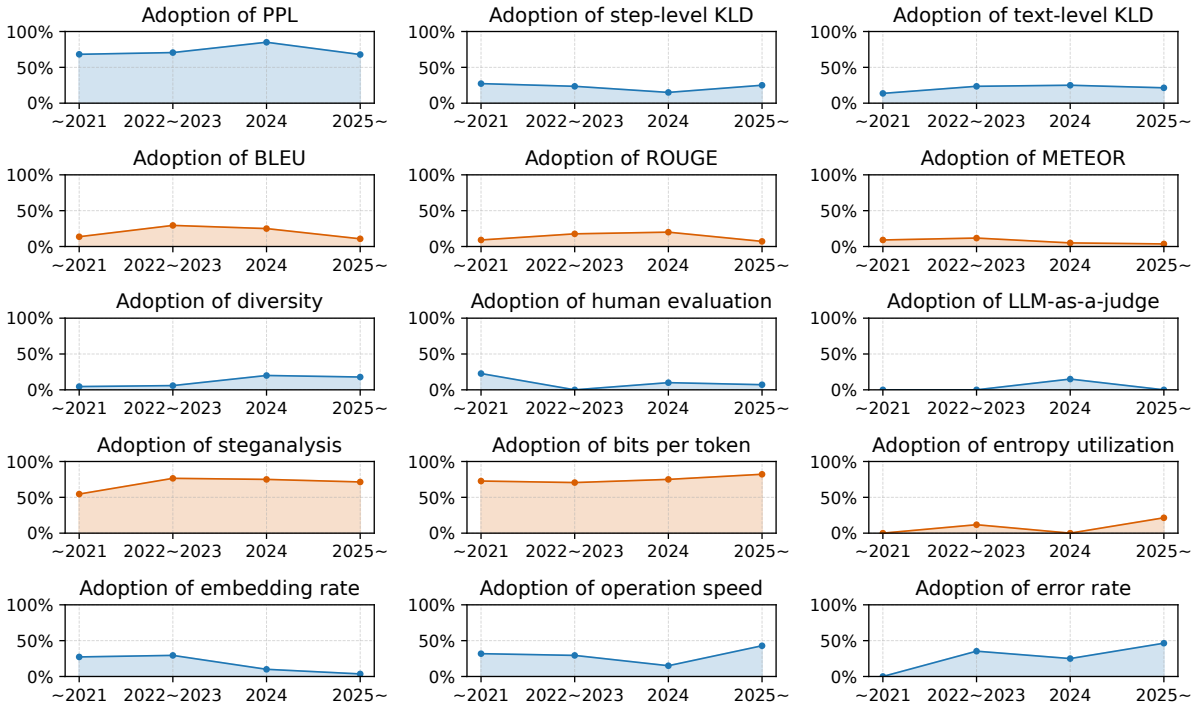


Figure 11: Adoption rates of the 15 major evaluation methods in 87 papers on modern generative linguistic steganography, presented in chronological order. Among these 87 papers, 22 were published before 2021, 17 between 2022 and 2023, 20 in 2024, and 28 after 2025.