A Semantic Controllable Long Text Steganography Framework Based on LLM Prompt Engineering and Knowledge Graph

Yihao Li, Ru Zhang, Jianyi Liu, *Member, IEEE*, and Qi Lei

***Abstract*—With ongoing advancements in natural language technology, text steganography has achieved notable progress. However, existing methods primarily concentrate on the probability distribution between words, often overlooking comprehensive control over text semantics. Particularly in the case of longer texts, these methods struggle to preserve coherence and contextual consistency, thereby increasing the risk of detection in practical applications. To effectively improve steganography security, we propose a semantic controllable long-text steganography framework based on prompt engineering and knowledge graph (KG) integration, obviating supplementary training. This framework leverages triplets from the KG and task descriptions to construct prompts, directing the large language model (LLM) to generate text that aligns with the triplet content. Subsequently, the model effectively embeds secret information by encoding the candidate pools established around the sampled target words. The experimental results demonstrate that our framework ensures the concealment of steganographic text while maintaining the relevance and consistency of the content as expected. Moreover, it can be flexibly adapted to various application scenarios, showcasing its potential and advantages in practical implementations.**

***Index Terms*—Text steganography, semantic controllable, LLM prompt engineering, knowledge graph.**

I. Introduction

Steganography is a technique of encoding secret information within common carriers (such as images [1], [2], [3], audio [4], [5], and text [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]), holding a significant position in the modern field of information security. This technology enables the secure transmission of secret information over insecure channels to the recipient without arousing the attention of potential eavesdroppers. Among various commonly used carriers, text steganography demonstrates higher practicality and concealment due to its minimal influence from transmission channels, thus attracting widespread research interest. Early text steganography primarily relied on modifications to the text, such as synonym substitution [6] and syntactic transformations [7]. However, these methods often tended to alter word frequencies, leading to poor concealment and susceptibility to detection by steganalysis techniques. To address this challenge, researchers turned to generation-based methods, which typically integrate language generation models based on probabilistic statistical distributions. Some studies [8], [9], [10], [11] initially explored schemes utilizing Markov chain models for embedding secret information. Due to the inherent limitations of Markov chain models, the concealment of such methods is often relatively weak.

Subsequently, researchers began exploring the use of neural networks as language generation models [12], [13], [14], [15], [16], [17]. Yang et al. [13] and Yang et al. [14] trained RNN and VAE on extensive corpora, then encoded the conditional probability distributions generated by the models, selecting corresponding words based on the secret information. These methods often focus solely on encoding words with the highest probabilities, potentially leading to heightened statistical disparities between the steganographic text and cover text. Particularly in applications involving long-text steganography, such differences can lead to noticeable performance degradation. Fang et al. [12] employed LSTM as a language generation model, dividing the text into several groups and pre-encoding them, then selecting the most probable words from the corresponding word blocks based on the secret information. Zhang et al. [15] proposed the provably secure ADG scheme, which adaptively divides words into several groups with approximately equal probabilities and selects a word from the corresponding group. However, steganographic texts generated by these methods are semantically uncontrollable, limiting their practicality.

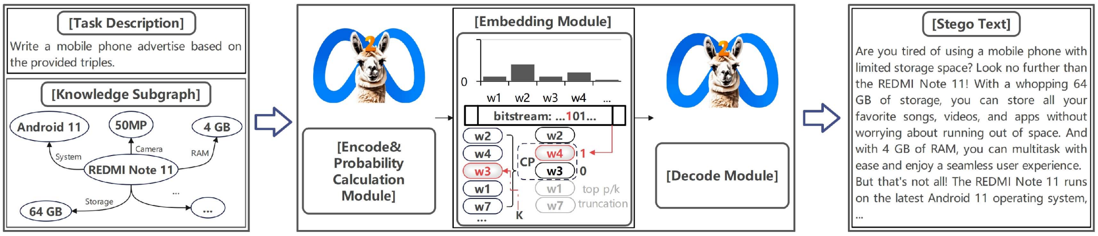
Recently, researchers have also begun to explore controllable semantic text steganography [18], [19], [20]. Li and Zhang et al. [18] proposed a scheme that achieves topic-controllable steganographic text generation by encoding relevant entities and their relationships in KG. Zhang et al. [19] employed an abstractive generation model and a dynamic programming-based embedding algorithm to ensure semantic consistency between the steganographic text and the target text. Despite achieving some success, these methods require secondary training on extensive corpora of target text before deployment. This not

Received 18 June 2024; revised 18 August 2024; accepted 24 August 2024. Date of publication 9 September 2024; date of current version 2 October 2024. This work was supported by the National Natural Science Foundation of China under Grant U21B2020. The associate editor coordinating the review of this article and approving it for publication was Dr. Yuanman Li. *(Corresponding author: Ru Zhang.)*

Yihao Li, Ru Zhang, and Jianyi Liu are with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: yihaoli@bupt.edu.cn; zhangru@bupt.edu.cn; liujy@bupt.edu.cn).

Qi Lei is with the 30th Research Institute of China Electronics Technology Group Corporation, Chengdu 610041, China (e-mail: rooster0428@126.com).

Digital Object Identifier 10.1109/LSP.2024.3456636


[Task Description]
Write a mobile phone advertise based on the provided triples.

[Stego Text]
Are you tired of using a mobile phone with limited storage space? Look no further than the REDMI Note 11! With a whopping 64 GB of storage, you can store all your favorite songs, videos, and apps without worrying about running out of space. And with 4 GB of RAM, you can multitask with ease and enjoy a seamless user experience. But that's not all! The REDMI Note 11 runs on the latest Android 11 operating system, ...

Fig. 1. The overall framework of our method. w3 is the target word sampled by the key K, w4 is the final output selected based on the secret bit, and CP represents the candidate pool constructed around w3 after applying the sampling restriction strategy of top-*p*/*k* truncation.

only consumes significant additional computational resources but also limits their transferability and flexibility since there may not always be sufficiently large corpora available for training in practical application scenarios. In contrast, prompt engineering [21] offers a highly flexible and customizable approach to text generation by directing the model to generate specific outputs through precise prompts. Additionally, knowledge graphs, consisting of triplets of entities and relationships, provide a rich form of structured knowledge representation [22]. Therefore, integrating prompt engineering with KG can aid LLMs in more effectively performing various downstream tasks, a pattern that has been applied across multiple domains such as knowledge reasoning [23], [24], [25], [26].

Based on the aforementioned facts, we propose a semantic-controllable long-text steganography framework without supplementary training requirements. This framework utilizes triplets from KG and task descriptions to construct prompts for LLM, allowing for precise semantic control of the steganographic text. This approach improves transferability across various application scenarios, thereby conserving substantial training resources. Different from previous strategies that directly encoded words based on maximum transition probabilities, we construct candidate pools around the target words sampled by the model. This allows for a meticulous search for the optimal positions to embed secret information, reducing the disparity between the embedding distribution and the actual distribution. As a result, this enhancement ensures higher quality and better concealment, while also showcasing its unique advantages in scenarios requiring precise semantic control.

II. The Proposed Method

The overall framework of our method is illustrated in Fig. 1. Specifically, during the text generation phase, we utilize prompts constructed from knowledge triplets and task descriptions to constrain the generated content. In the information hiding phase, we construct candidate pools around the target words sampled by the model to embed the secret information, thereby enhancing the concealment of the steganographic text. In the information extraction phase, we decode the received text to retrieve the secret information.

A. Knowledge Graph-Guided Text Generation

Initially, we define a text generator LLM and a KG $G = \{(e_1, r, e_2) \mid e_1, e_2 \in E_G,\ r \in R_G\}$, where E_G and R_G respectively represent the entity set and relation set. In accordance with the specific requirements of the task, selected triplets from G are utilized to construct a knowledge subgraph G_s, as delineated in (1).

$$G_s = \{(e_1, r, e_2) \in G \mid (e_1, r, e_2) \text{ are relevant to } D\} \tag{1}$$

Leveraging both G_s and the task description D, the LLM is directed to generate semantically specific text C_{content}, as formalized in (2). And C_{content} can be further expressed as a sequence of words $\{c_1, c_2, \ldots, c_{\text{end}}\}\quad(0 < \text{end} < \text{max\_length})$, where max_length is the maximum text length allowed by the LLM. For this research, we use Llama 7B Chat [27] as the text generator.

$$C_{\text{content}} = \text{Gen}(LLM \mid G_s, D) \tag{2}$$

B. Information Hiding

In contrast to conventional methodologies, we adopt a different strategy to construct the candidate pool. Typically, most previous practices involve encoding only the top 2^n words with the highest probabilities. Instead, we first arrange the probability distribution $p(W_t)$ in descending order and perform random sampling to determine the sampling target word at time step t. To ensure the reproducibility of this sampling process during decoding, we refer to the principle of symmetric encryption to pre-specify a random number generation seed as the key k, which must be shared between the sender and the receiver. The process of sampling the target word $c_{\text{t\_target}}$ according to the key k is illustrated below:

$$c_{\text{t\_target}}, i, p(W_t') = \text{Sampling}(k, p(W_t)) \tag{3}$$

where i represents the index of $c_{\text{t\_target}}$ within W_t' and $p(W_t')$ represents the probability distribution sorted in descending order.

Then we select the words adjacent to the target word $c_{\text{t\_target}}$ from the probability distribution $p(W_t')$ to construct the candidate pool CP_t, as delineated in (4).

$$CP_t = \left\{ w_j \in W_t' \mid i - \frac{c}{2} \le j < i + \frac{c}{2} \right\} \tag{4}$$

Finally, the words in the CP_t are encoded, denoted by E_t, and the corresponding words are selected based on the secret bits to obtain the final output word s_t, as shown in (5)–(6). And the final steganographic text S_{content} can be further expressed as a sequence of words, as illustrated in (7). Drawing from previous research, we employ two different encoding methods: Huffman coding [13] and arithmetic coding [14].

$$E_t{=}\text{Encode}(CP_t){=}\{e_{t1}, e_{t2}, \ldots, e_{tc}\} \tag{5}$$

$$s_t{=}\text{Match}(m, E_t){=}\{w_{ti} \in CP_t \,|\, m \text{ matches } e_{ti}, \forall e_{ti} \in E_t\} \tag{6}$$

$$S_{\text{content}}{=}\bigcup_{i=0}^{\text{end}} \text{Match}(m, E_i){=}\{s_1, s_2, \ldots, s_{\text{end}}\} \tag{7}$$

Additionally, to prevent including words with low probabilities in the candidate pool and thus affecting the quality of the generated text, we employ sampling limit strategies such as top-p and top-k to further limit the construction of CP. Therefore, in practical applications, the size of the candidate pool constructed at each time step is $c' = 2^b \quad (0 \leq b \leq \log c)$. When $b = 0$, it indicates that no secret information is embedded at this time step.

C. Information Extraction

Upon receiving the steganographic text, the receiver needs to correctly decode the hidden secret information. Information extraction is the reverse process of information hiding, we firstly utilize the same language model LLM, knowledge subgraph G_s, task description D, and key k to compute the conditional probability distribution of each word and reconstruct the candidate pool at each time step, which is also consistent with (3)–(5). Then, we use the same encoding method to encode the received words and extract the embedded secret bits m based on the received text. The entire process is represented by (8)–(9), where Ex $(\cdot)$ denotes the process of extracting secret bits based on the actual received words.

$$\text{Ex}(r, E_t) = \{e_{ti} \in E_t \mid w_{ti} \text{ matches } r, \forall w_{ti} \in CP_t\} \tag{8}$$

$$m = \bigcup_{s_i \in S_{\text{content}}} \text{Ex}(s_i, E_i) \tag{9}$$

III. Experiments and Analysis

A. Setup

1) Dataset and Model Configuration: Experiments were conducted on three datasets: Story, Post, and Ad. The Story dataset was generated by ChatGPT [28], while Post and Ad were created by ChatGPT using the Recipe Dataset and Mobile Recommendation System Dataset from Kaggle, respectively. Each dataset entry includes multiple triplets detailing a story, recipe, or mobile phone. Our text generation tasks involved generating narrative stories, recipe blogs, and phone advertisements using Meta LLaMA2 7B Chat. We employed two encoding algorithms (arithmetic coding and Huffman coding), two candidate pool sizes (c = 4 and c = 8), and two LLM sampling strategies (top-p = 0.9 and top-k = 100).

2) Baselines: We selected five state-of-the-art (SOTA) methods as baselines: Bin [12], ADG [15], HC [13], AC [14], and Discop [31]. To determine the next token, Bin divides words into 2^B blocks, ADG adaptively partitions the vocabulary into roughly equal probability groups, HC constructs a Huffman tree using the top 2^H most probable words, AC encodes the top K most probable words using arithmetic coding and Discop constructs multiple "distribution copies". Since these methods require extensive pre-training on the target corpus, we uniformly used LLaMA2 7B Chat as the text generator for this experiment.

TABLE I
Embedding Quantities Comparison Between Our Method and Baselines Under Different Settings

Type	Method	Story		Post		Ad	
		ER	bpm	ER	bpm	ER	bpm
block	ADG	0.7052^1	310.86	0.4328^1	209.57	0.2756^1	125.01
	Bin (B=1)	1^2	399.30	1^2	497.12	1^2	408.09
-	Discop (*p*=0.9)	0.8216^5	287.31	0.3433^5	146.71	0.4683^5	189.21
AC	AC (k=100)	0.9272^3	379.77	0.6693^3	303.66	0.5276^3	238.67
	ours_AC (c=4)-*p*	0.3647	142.40	0.1831	84.27	0.1729	70.67
	ours_AC (c=8)-*p*	0.4140	168.85	0.3105	139.86	0.1851	77.05
	ours_AC (c=4)-*k*	0.6817^1	283.51	0.4072^1	194.29	0.3596	151.61
	ours_AC (c=8)-*k*	0.9474^{35}	414.93	0.8260^{23}	374.22	0.4505^{35}	197.26
HC	HC (H=1)	1^4	362.88	1^4	504.89	1^4	388.64
	ours_HC (c=4)-*p*	0.5748	224.51	0.3248	151.63	0.3138^1	124.72
	ours_HC (c=8)-*p*	0.6239	250.93	0.3575^5	159.80	0.3253	130.05
	ours_HC (c=4)-*k*	0.9480	422.90	0.8100	415.99	1.0006^{24}	393.86
	ours_HC (c=8)-*k*	0.9576^{24}	406.12	0.8169^4	396.85	0.9607	388.91

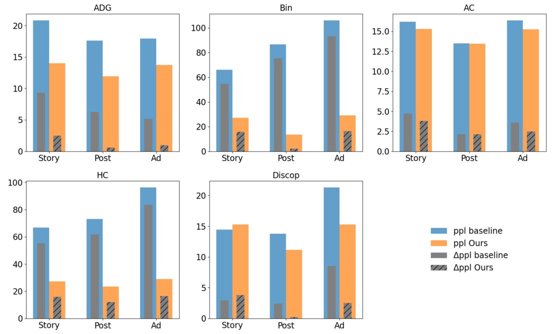


Fig. 2. Comparison of text quality between our method and baseline methods under similar ER conditions.

B. Results

1) Payload Embedding: We calculated the embedding rate (ER) and the average number of secret bits per message (bpm), as shown in Table I. In long-text steganography, a single word carrying one secret bit can embed a significant amount of secret information. Therefore, it is crucial to find an optimal balance between text quality and embedding quantity in practical applications. Accordingly, our experiments focused exclusively on ER within the [0, 1] range.

In general, our method exhibits increased ER as the candidate pool size increases. Additionally, under sampling constraints, ER was slightly higher with top-$k = 100$ than with top-$p = 0.9$. This may be due to the stricter top-p = 0.9 constraint, which reduced the number of eligible words and, consequently, the candidate pool size in our experiments. The same superscript in Table I denotes settings within our method that have similar ER to the baselines, which are intended for subsequent experimental comparisons.

2) Text Fluency: The evaluation of steganographic text fluency is presented in Figs. 2 and 3. Overall, as the embedding rate increases, the naturalness of the text generated by our method tends to decrease. Nevertheless, our method demonstrates strong

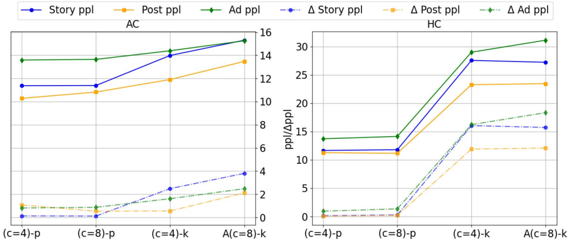


Fig. 3. Comparison of text quality using our method under different ER conditions.

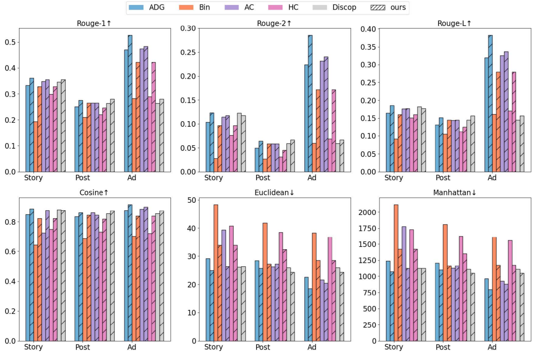


Fig. 4. Comparison of semantic consistency between our method and baseline methods under similar ER conditions.

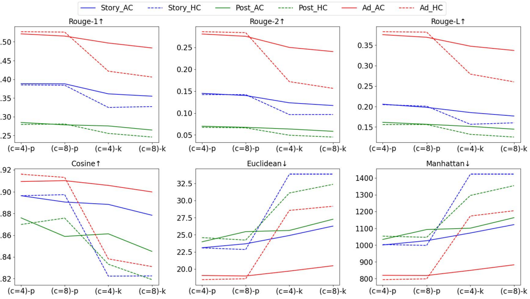


Fig. 5. Comparison of semantic consistency using our method under different ER conditions.

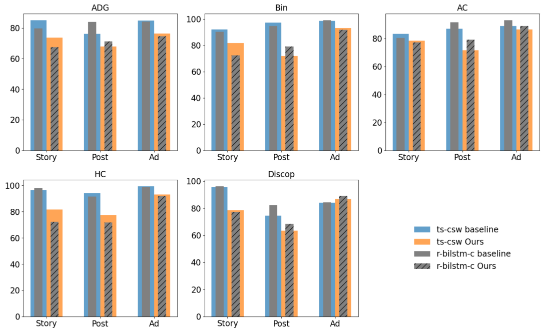


Fig. 6. Comparison of detection accuracy scores between our method and baseline methods using TS-CSW and R-BiLSTM-C under similar ER conditions.

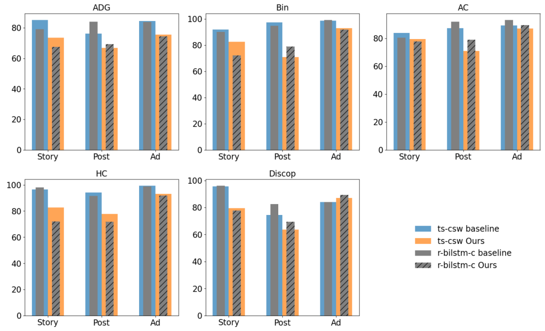


Fig. 7. Comparison of detection F1 scores between our method and baseline methods using TS-CSW and R-BiLSTM-C under similar ER conditions.

perceptual concealment, outperforming baselines in almost all experiments. In particular, our method achieves a reduction in *ppl* exceeding 23% and in Δ*ppl* over 72% compared to ADG, HC and Bin across all three datasets. In comparison with Discop, our method achieves a decrease in *ppl* of over 18% and in Δ*ppl* of more than 70% on the Post and Ad datasets, though its performance on the Story dataset was slightly inferior.

3) Semantic Consistency and Transferability: In Fig. 5, it can be observed that with the rise in candidate pool size and embedding rate, our method may experience a decrease in performance due to the selection of words that are "farther" from the original target words. However, our method outperforms the baselines in almost all metrics, generating steganographic text with closer semantic alignment to the reference text, as shown in Fig. 4. Additionally, the application of our framework across different tasks underscores its adaptability and transferability.

4) Anti-Steganalysis Ability: We adopted steganalysis methods: TS-CSW [29] and R-BiLSTM-C [30]. For each dataset, we generated 500 steganographic texts and 500 non-steganographic texts, which were then divided into training and test sets in an 8:2 ratio. From Figs. 6 and 7, it can be observed that our method exhibits lower detection accuracy compared to baselines. This indicates that our method demonstrates stronger resistance to steganalysis and hiding abilities. Particularly, in both detection algorithms on the Post dataset, our method shows a decrease in detection accuracy of over 10% compared to all baselines. This improvement can be attributed to the construction of candidate pools around the target words, which aids in avoiding significant statistical differences between the steganographic text and cover text, thereby reducing detectability.

IV. Conclusion

In this letter, we propose a semantic controllable long-text steganography framework, aiming to address issues such as uncontrollable text content and additional training resource consumption in traditional text steganography methods. Experimental results demonstrate that our proposed method ensures content consistency between the steganographic text and the target text while also guaranteeing superior text quality and resistance to detection compared to baselines, providing a novel solution for semantic controllable text steganography. However, although the method is adaptable to various tasks, its transferability is primarily limited to textual content. Generating content with specific linguistic habits or uncommon styles may still require additional fine-tuning.

References

[1] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, “Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research,” *Neurocomputing*, vol. 335, pp. 299–326, 2019.

[2] Y. Luo, J. Qin, X. Xiang, and Y. Tan, “Coverless image steganography based on multi-object recognition,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 7, pp. 2779–2791, Jul. 2021.

[3] D. Zhang, X. Chen, F. Li, A. K. Sangaiah, and X. Ding, “Seam-Carved image tampering detection based on the cooccurrence of adjacent LBPs,” *Secur. Commun. Netw.*, vol. 2020, pp. 1–12, Dec. 2020.

[4] A. A. AlSabhany, A. H. Ali, F. Ridzuan, A. H. Azni, and M. R. Mokhtar, “Digital audio steganography: Systematic review, classification, and analysis of the current state of the art,” *Comput. Sci. Rev.*, vol. 38, 2020, Art. no. 100316, ISSN 1574-0137.

[5] Z. Yang, X. Peng, Y. Huang, and C. Chang, “A novel method of speech information hiding based on 3D-Magic matrix,” *J. Internet Technol.*, vol. 20, no. 4, pp. 1167–1175, 2019.

[6] L. Huo and Y. Xiao, “Synonym substitution-based steganographic algorithm with vector distance of two-gram dependency collocations,” in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, 2016, pp. 2776–2780.

[7] M. Kim, O. Zaiane, and R. Goebel, “Natural language watermarking based on syntactic displacement and morphological division,” in *Proc. 34th Annu. IEEE Comput. Softw. Appl. Conf. Workshops*, 2010, pp. 164–169.

[8] H. H. Moraldo, “An approach for text steganography based on Markov chains,” 2014, *arXiv:1409.0915*.

[9] A. N. Shniperov and K. A. Nikitina, “A text steganography method based on markov chains,” *Aut. Control Comp. Sci.*, vol. 50, pp. 802–808, 2016.

[10] Y. Dai, W. Dai, Y. Yu, and B. Deng, “Text steganography system using Markov chain source model and DES algorithm,” *J. Softw.*, vol. 5, no. 7, pp. 785–792, 2010.

[11] Y. Luo, Y. Huang, F. Li, and C. Chang, “Text steganography based on ci-poetry generation using Markov chain model,” *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 9, pp. 4568–4584, 2016.

[12] T. Fang, M. Jaggi, and K. Argyraki, “Generating steganographic text with LSTMs,” in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics - Student Res. Workshop*, 2017, pp. 100–106.

[13] Z. Yang, X. Guo, Z. Chen, Y. Huang, and Y. Zhang, “RNN-stega: Linguistic steganography based on recurrent neural networks,” *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 5, pp. 1280–1295, May 2019.

[14] Z. Yang, S. Zhang, Y. Hu, Z. Hu, and Y. Huang, “VAE-Stega: Linguistic steganography based on variational auto-encoder,” *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 880–895, 2021.

[15] S. Zhang, Z. Yang, J. Yang, and Y. Huang, “Provably secure generative linguistic steganography,” in *Proc. Int. Joint Conf. Natural Lang. Process. (ACL-IJCNLP)*, 2021, pp. 3046–3055.

[16] T. Lu, G. Liu, R. Zhang, and T. Ju, “Neural linguistic steganography with controllable security,” in *Proc. 2023 Int. Joint Conf. Neural Netw. (IJCNN)*, 2023, pp. 1–8.

[17] H. Wang, Z. Yang, J. Yang, Y. Gao, and Y. Huang, “Hi-stega: A hierarchical linguistic steganography framework combining retrieval and generation,” in *Proc. Int. Conf. Neural Inf. Process. (ICONIP)*, 2023, pp. 41–54.

[18] Y. Li, J. Zhang, Z. Yang, and R. Zhang, “Topic-aware neural linguistic steganography based on knowledge graphs,” *ACM/IMS Trans. Data Sci.*, vol. 2, no. 2, pp. 1–13, 2021.

[19] R. Zhang, J. Liu, and R. Zhang, “Controllable semantic linguistic steganography via summarization generation,” in *Proc. 2024 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2024, pp. 4560–4564.

[20] T. Yang, H. Wu, B. Yi, G. Feng, and X. Zhang, “Semantic-preserving linguistic steganography by pivot translation and semantic-aware bins coding,” *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 1, pp. 139–152, Jan./Feb. 2024.

[21] Q. Ye, M. Axmed, R. Pryzant, and F. Khani, “Prompt engineering a prompt engineer,” in *Findings of the Association for Computational Linguistics: ACL 2024*, Bangkok, Thailand, 2024, pp. 355–385.

[22] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, “Unifying large language models and knowledge graphs: A roadmap,” *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3580–3599, Jul. 2024.

[23] J. Baek, A. F. Aji, and A. Saffari, “Knowledge-augmented language model prompting for zero-shot knowledge graph question answering,” in *Proc. 1st Workshop Natural Lang. Reasoning Structured Explanations (NLRSE)*, 2023, pp. 78–106.

[24] J. Jiang, K. Zhou, Z. Dong, K. Ye, X. Zhao, and J.-R. Wen, “StructGPT: A. general framework for large language model to reason over structured data,” in *Proc. 2023 Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 9237–9251.

[25] X. Li et al., “Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources,” in *Proc. 12th Int. Conf. Learn. Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=cPgh4gWZlz

[26] J. Wang, Q. Sun, X. Li, and M. Gao, “Boosting language models reasoning with chain-of-knowledge prompting,” in *Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand, 2024, pp. 4958–4981.

[27] H. Touvron et al., “Llama 2: Open foundation and fine-tuned chat models,” 2023, *arXiv:2307.09288*.

[28] OpenAI et al., “GPT-4 technical report,” 2023, *arXiv:2303.08774*.

[29] Z. Yang, Y. Huang, and Y. J. Zhang, “TS-CSW: Text steganalysis and hidden capacity estimation based on convolutional sliding windows,” *Multimedia Tools Appl.*, vol. 79, pp. 18293–18316, 2020.

[30] Y. Niu, J. Wen, P. Zhong, and Y. Xue, “A hybrid R-BILSTM-C neural network based text steganalysis,” *IEEE Signal Process. Lett.*, vol. 26, no. 12, pp. 1907–1911, Dec. 2019.

[31] J. Ding, K. Chen, Y. Wang, N. Zhao, W. Zhang, and N. Yu, “Discop: Provably secure steganography in practice based on ‘distribution copies’,” in *Proc. IEEE Symp. Secur. Privacy*, 2023, pp. 2238–2255.